

ELBUG

FOR THE ELECTRON

George and the Dragon

Vol 2 No 2 December 1984

GAMES

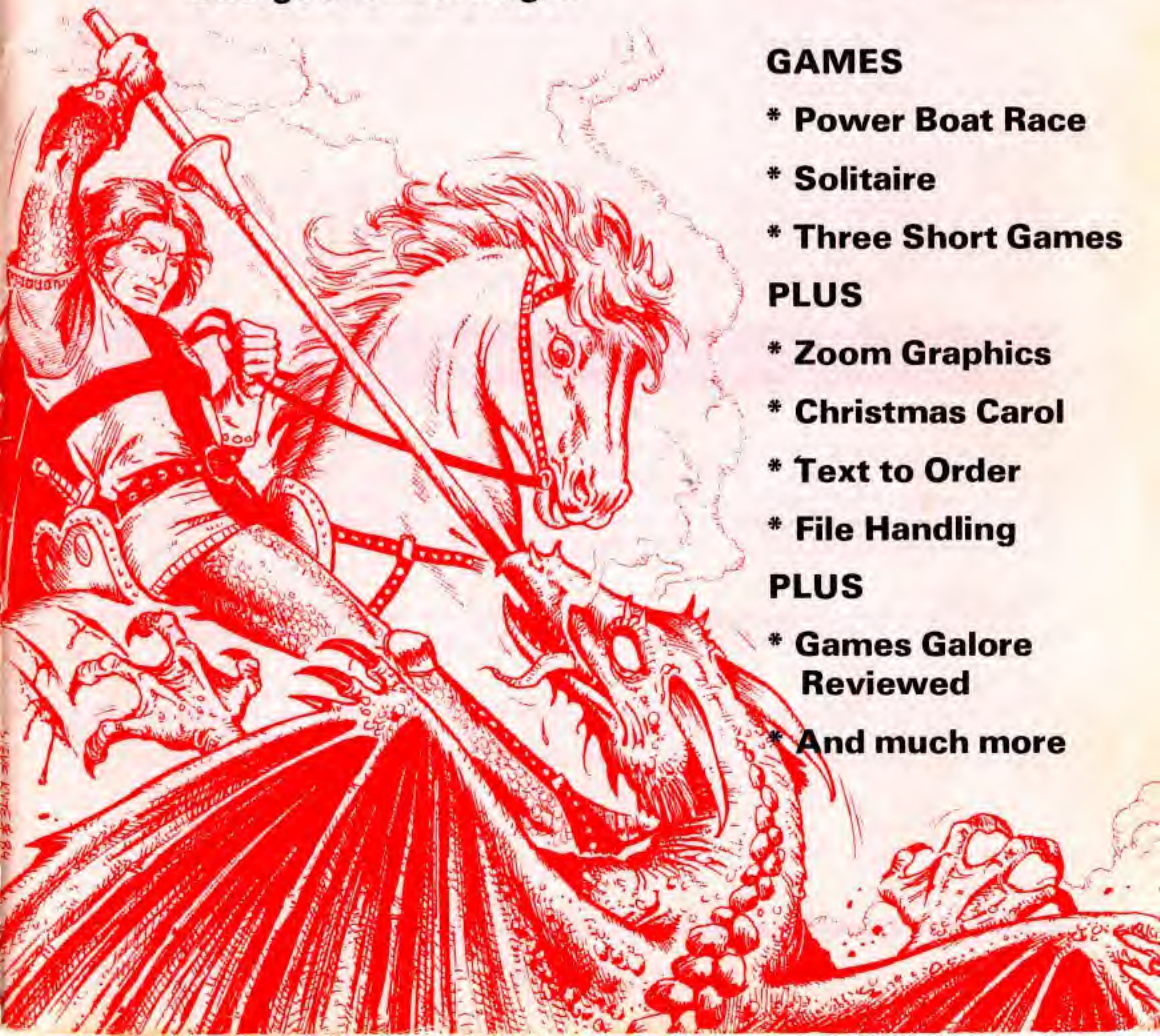
- * **Power Boat Race**
- * **Solitaire**
- * **Three Short Games**

PLUS

- * **Zoom Graphics**
- * **Christmas Carol**
- * **Text to Order**
- * **File Handling**

PLUS

- * **Games Galore Reviewed**
- * **And much more**



EDITORIAL

CHRISTMAS ISSUE

This is the Christmas issue of ELBUG with a bumper crop of reviews to help with last minute present buying, and more games than ever to keep you amused over Christmas and the New Year. In addition to our usual three excellent full-length games, we have also included a further three very short but no less challenging games for your entertainment. There is also a rendering of a popular Christmas carol together with its own fascinating graphics display.

COMPETITION

We think that the Zoom Graphics program published this month is so fascinating that we have organised a competition for you as well, though for technical reasons this is only feasible on the magazine cassette, and not printable in the magazine itself. If you don't already subscribe to the magazine cassette, why not send off today (see back cover for details) and give yourself a chance of winning £25 of Electron software. Watch out for another interesting competition in the next issue of ELBUG.

ELECTRON USER GUIDE INDEX

As well as the index to volume one of ELBUG, we were also able to include with your magazine last month a further additional item in the form of an index to the Electron User Guide. We hope you will find this really useful. While still available, this index will also be sent out to new members with their first issue of ELBUG.

ELECTRON BLOSSOMS

Acorn now claim to have sold 170,000 Electrons and you may well have seen the TV advertising campaign aimed at making the Electron one of the best selling micros this Christmas. At a recent computer exhibition, the new PLUS 3 disc unit was to be seen in operation, together with Electrons using second processors and serial interfaces. Acorn clearly see a promising future for the Electron.

Mike Williams

NOTICE BOARD NOTICE BOARD NOTICE BOARD NOTICE BOARD

WEE SHUGGY

Unfortunately, the superb game Wee Shuggy in the last issue (Vol.2 No.1) was marred by the omission of spaces in lines 290 and 300. These should be inserted between keywords and variable names in these three lines when typing in the program. This arose because the version used had been previously compacted by the author. The compaction causes no problem on the magazine cassette version.

HINT WINNERS

This month the £10 prize goes to Peter Sandford and the £5 prize to M.R.Bowers. We are always interested in hearing of your hints and tips for Electron users.

MAGAZINE CASSETTE

This month the magazine cassette contains a total of nine programs from the magazine together with the data file for use with the Zoom program and the Zoom competition (see the back cover for full ordering details).

ELBUG MAGAZINE

GENERAL CONTENTS

Page Contents

2	Editorial
4	Zoom
7	News
8	Power Boat Race
12	Alligata's Contract Bridge Reviewed
13	Christmas Carol
14	Games Galore
18	One, Two, Three Games for Christmas
20	Text to Order
23	White Knight Reviewed
24	Solitaire
27	File Handling (Part 1)
30	George and the Dragon

HINTS, TIPS & INFO

7	Rounding Numbers
19	Macros in Assembler
22	Even Better Input?
22	Noise for Fireworks
22	Tidy Formatting
26	Decimals to Fractions
29	Basic Memory Reserve
29	More Memory
29	Strange Error Lines
33	Better Input
33	New Way to Crash an Electron
33	Strange Variables

PROGRAMS

4	Zoom Graphics
8	Power Boat Race Game
13	Christmas Carol
18	Three Short Games
20	Text to Order
24	Solitaire
27	File Handling Example Programs
30	George and the Dragon Game

ZOOM

by Pete and Derek Chowns

For such a short program this one is remarkably entertaining. It allows you to zoom in and out of any picture you create with magnifications of up to 10 million! It is based on a perfectly serious application in Computer-Aided Design (CAD), but can be used to construct line drawings of high precision, or light-hearted sketches containing microscopic detail for a game of "Hide and Seek".

INTRODUCTION

You may remember from the BBC TV series, "The Computer Programme", the demonstration in which a house was shown with a globe in the window. The picture was magnified to show the British Isles on the globe. On the map was a house. In the window of the house was a globe... All that is possible on the Electron with this little program. If you have a printer with graphics capability, you can add your own routine to print the results (though this would also require a printer interface for your Electron).

The 'ZOOM' program allows you to produce line drawings, and at any stage, enlarge or reduce the size of the picture. This allows details to be drawn much larger than their final size for accurate results, and also allows part of a picture to be hidden, by reducing it so small that it is no longer visible on the screen. The program is entirely in Basic, and should be quite straightforward to type in and save on cassette.

USING THE ZOOM PROGRAM

Only six commands plus the four cursor-control (arrow) keys and the Shift key are used to achieve the pictures accompanying this article. When you run the program a small dot will appear in the centre of the screen. This is the position of the

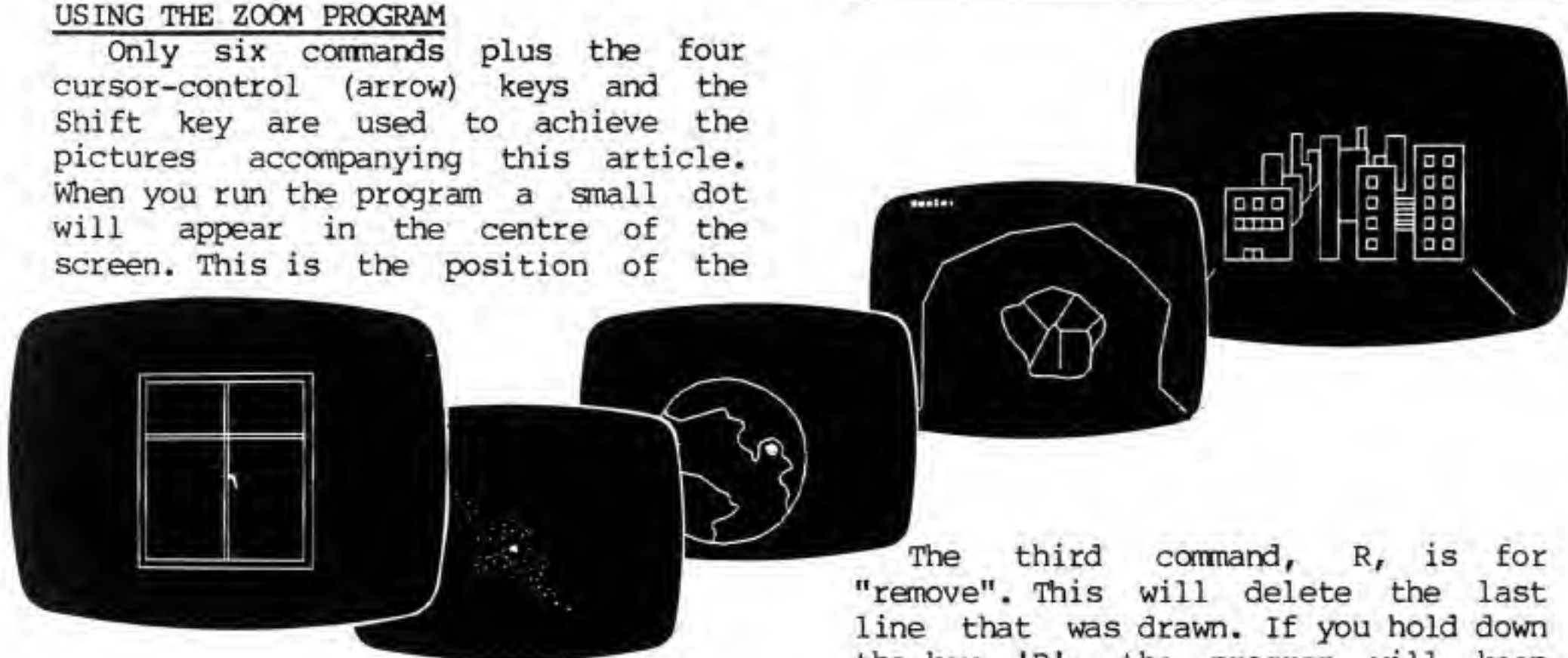
cursor. Hold down one of the cursor keys and a line will be drawn (holding down Shift at the same time moves the cursor much faster). Use different cursor keys and the line can be moved as if connected to the starting point by a piece of elastic.

When the line is where you want it, press D for "draw" and the line becomes fixed. If you merely wish to move the starting point of the line without drawing it, press M for "move". So the procedure is: stretch the elastic to a new position, and then press either D or M.

ZOOM Key Functions

M - Move
D - Draw
R - Remove
S - Scale
P - Print
Func/f0 - Save screen
Func/f1 - Load screen

Cursor keys move cursor



The third command, R, is for "remove". This will delete the last line that was drawn. If you hold down the key 'R', the program will keep

deleting lines, until all the lines have been removed. If removal of a line leaves a 'hole' in any remaining line, then the drawing can be 'repaired' by re-drawing with a scale of 1 (see later).

The fourth command is S for "scale". Type S and the word "Scale:" appears in the top left corner of the screen. Then type in a positive number for the magnification you require, e.g. 2 = twice as big, 0.5 = half as big, 1 = same size. When Return is pressed the picture is re-drawn centred on the latest position of the cursor and scaled by the factor specified.

The command, P for "print", is only useful if you have a graphics dump routine, a printer and printer port. Press 'P' at any time to print the current screen display. The program listing shows where a printer dump routine (or command to one of the new printer dumps like *SCREENDUMP which is supplied with the SIR expansion) should be inserted.

We have also incorporated a simple facility to save and load screen displays, using the two function keys f0 for save and f1 for load. Remember that you have to press Func with either of these to get the correct effect. In each case the program will ask you for the filename to save or load. Be careful when saving and loading as pressing f1 instead of f0 will delete the current picture in memory.

ZOOM IN PRACTICE

Some useful features of this program are the ability to create precise drawings, by building your picture up on a large scale, and then shrinking it down to size. You can also translate your shape in any of four directions. For example, if you have a square in the middle of the screen, then move the cursor to the top right-hand corner of this square. The press 'S' and choose a scale of 1 (same size). The square will now be re-drawn with the top right-hand corner in the middle of the screen.

LIMITATIONS

The program at present will allow drawings to be created with up to 600 points, set in the program at line 110. This could be increased and the

ultimate limit is like to be determined by the amount of memory available for storing the lengthy arrays declared at line 270.

Using the scaling feature actually changes the stored data values. If you magnify any part of your drawing by more than about 10 million, you will be exceeding the accuracy of the micro and the results may be unrecognisable. You may also find that because the stored data has been changed, you cannot reduce the picture correctly back to the original version. A screen will always be saved at the current magnification.

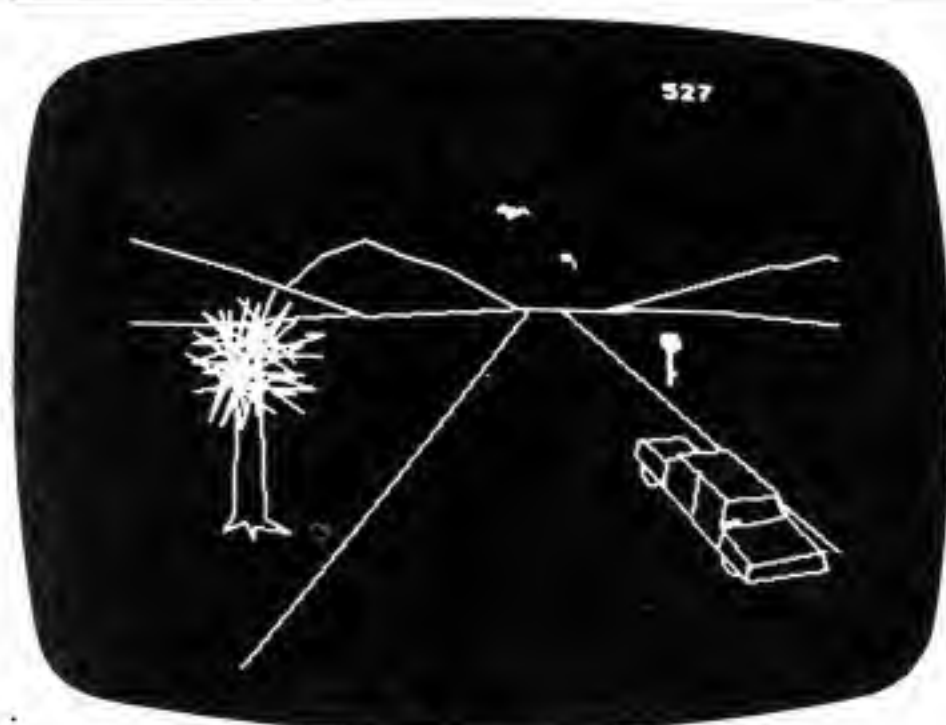
ZOOM FOR FUN

The program really is great fun to use, and you can puzzle your friends by hiding some minute detail somewhere in a picture and challenging them to find what it is. If you try this, we have found that it is better to start at the highest level and gradually magnify parts of the picture as you get more and more detailed. If you start with the smallest details and gradually reduce these to hide them, you are likely to find a very obvious 'blob' being drawn as part of the initial picture, a real give away to the location of the hidden detail.

The pictures with this article show one such sequence. The first shows a window with a view of the stars in the sky (actually the constellation of Orion). If you zoom in on one of the stars in Orion you will see a planet surrounded by asteroids. Magnifying the central planet will reveal the outline of some continents and an island, and further magnification of the island will show first the road system on the island, and finally a city of tall buildings. You soon realise that you only have to be as little as a millimetre out initially to be a 100 kilometres adrift at a magnification of 10 million.

ZOOM TREASURE HUNT

Because ZOOM is such an interesting program we have provided a data file, DZOOM, on the magazine cassette for you to explore. Unfortunately the data file is much too large and in the wrong format to be included in the magazine



itself. If you search hard enough you will find the name 'ELBUG' hidden somewhere in the picture you see here. Tell us where 'ELBUG' is located and we will award a prize of £25 of Electron Software to the first correct answer opened. Send all entries to the editorial address, and mark the envelope 'ZOOM'. Good zooming!

```

10 REM PROGRAM ZOOM
20 REM VERSION E0.1
30 REM AUTHORS P.CHOWN A.R.WEBSTER
40 REM ELBUG DECEMBER 1984
50 REM PROGRAM SUBJECT TO COPYRIGHT
60 :
100 ON ERROR GOTO 390
110 np%=600
120 *FX4,2
130 *K.0 z
140 *K.1 y
150 *K.10|UO.|M
160 *K.12|H
170 *K.13|I
180 *K.14|J
190 *K.15|K
200 *FX12,3
210 :
220 MODE 4
230 VDU19,1,2,0,0,0
240 X=639.5:Y=511.5
250 oldX=X:oldY=Y
260 pnt%=0
270 DIM endx(np%),endy(np%),startx(np
%),starty(np%)
280 GCOL 4,0
290 PROCline
300 REPEAT
310 VDU4:PRINTTAB(30,0);pnt%;SPC(5):V
DU5
320 COM$=GET$: *FX15,0
330 IF COM$="P" PROCdump
340 PROCline

```

```

350 PROCcom
360 PROCline
370 UNTIL 0
380 :
390 ON ERROR OFF
400 *FX4
410 *FX12
420 MODE 6
430 IF ERR=17 END
440 REPORT:PRINT" at line ";ERL
450 END
460 :
1000 DEF PROCline
1010 MOVE oldX,oldY
1020 *FX19
1030 PLOT21,X,Y
1040 ENDPROC
1050 :
1060 DEF PROCcom
1070 IF INKEY-1 C=32 ELSE C=4
1080 IF COM$=CHR$(8) OR COM$=CHR$(140)
X=X-C
1090 IF COM$=CHR$(9) OR COM$=CHR$(141)
X=X+C
1100 IF COM$=CHR$(10) OR COM$=CHR$(142)
Y=Y-C
1110 IF COM$=CHR$(11) OR COM$=CHR$(143)
Y=Y+C
1120 IF COM$="M" oldX=X:oldY=Y
1130 IF COM$="D" GCOL 0,1:MOVE X,Y:DRA
W oldX,oldY:GCOL 4,0:startx(pnt%)=oldX:
starty(pnt%)=oldY:endx(pnt%)=X:endy(pnt
%)=Y:pnt%=pnt%+1:oldX=X:oldY=Y
1140 IF COM$="R" AND pnt%<>0 pnt%=pnt%
-1:GCOL0,0:MOVE startx(pnt%),starty(pnt
%):DRAW endx(pnt%),endy(pnt%):GCOL4,0
1150 IF COM$="S" AND pnt%>0 PROCscale(
1)
1160 IF COM$="z" OR COM$="y" PROCsavel
oad
1170 ENDPROC
1180 :
1190 DEF PROCscale(sc)
1200 IF sc=0 GOTO 1240 ELSE MOVE 0,1023
1210 VDU5:GCOL 4,0:INPUT "Scale:"SCALE$
1220 SCALE=VAL(SCALE$)
1230 IF SCALE<=0 VDU 7:ENDPROC
1240 CLS:GCOL 0,1
1250 FOR I%=0 TO pnt%-1
1260 startx(I%)=(startx(I%)-X)*SCALE+6
39.5:starty(I%)=(starty(I%)-Y)*SCALE+51
1.5
1270 endx(I%)=(endx(I%)-X)*SCALE+639.5
:endy(I%)=(endy(I%)-Y)*SCALE+511.5
1280 IF ABS(startx(I%))<32768 AND ABS(
starty(I%))<32768 AND ABS(endx(I%))<327
68 AND ABS(endy(I%))<32768 MOVE startx(
I%),starty(I%):DRAW endx(I%),endy(I%)
1290 NEXT
1300 X=639.5:Y=511.5:oldX=X:oldY=Y

```

```

1310 GCOL 4,0:VDU 7
1320 ENDPROC
1330 :
1340 DEFPROCdump
1350 REM VDU5:*SCREENDUMP
1360 ENDPROC
1370 :
1380 DEFPROCsaveLoad
1390 VDU4:VDU31,0,0
1400 IF COM$="y" PRINT " Load "; ELSE
PRINT " Save ";
1410 INPUT"Filename : "fn$:CLS:VDU5
1420 IF COM$="y" GOTO 1500
1430 X=OPENOUT fn$
1440 PRINT #X,pnt%
1450 FORp=0TOpnt%

```

```

1460 PRINT #X,startx(p),starty(p),endx
(p),endy(p)
1470 NEXT
1480 CLOSE#0
1490 GOTO 1570
1500 :
1510 X=OPENUP fn$
1520 INPUT #X,pnt%
1530 FORp=0TOpnt%
1540 INPUT #X,startx(p),starty(p),endx
(p),endy(p)
1550 NEXT
1560 CLOSE#0
1570 SCALE=1:X=640:Y=512:PROCscale(0):
*FX15
1580 ENDPROC

```

NEWS NEWS NEWS NEWS

STAND UP AND BE COUNTED

Acorn now claims that over 90,000 Electrons have been sold and that all production difficulties are over. Such numbers can only encourage more software and add-ons to appear.

NO MODE 7 AFTER ALL

Sir Computers, manufacturers of the ROM board and Printstick Electron add-ons, have gone into voluntary liquidation. Although there are sufficient other manufacturers of joystick, printer, and ROM interfaces for the Electron, the sad loss to Electron owners will be the forthcoming Mode 7 adaptor that Sir was to have completed this month. This will now, of course, not appear.

THE ELECTRON VIEWPOINT

View, Acornsoft's wordprocessor, is now available for the Electron. View comes as a Rom cartridge for owners of the Plus-1 interface only at a price of about £49. View has been available on

the BBC micro for some time now and is also to be bundled with Acorn's ABC business systems when they appear. To complement View, the newer Viewsheet spreadsheet package is also being made available as an Electron cartridge.

Another cartridge from Acorn, to grace your Plus-1 socket is the RS232 interface. This will allow you to use RS232 printers, modems and other such serial goodies with your machine.

PLUS-3

The Plus-3 disc drive add-on looks set to appear just in time for Christmas. Although displayed at a couple of shows, Acorn's Plus-3 has not yet made it to the shops. Acorn claims that the unit will be available in quantity by Christmas, though there are no details as to price as yet. The Plus-3 will add a single 320K 3.5 inch disc drive to the Electron along with all the joys of fast and reliable data storage that BBC disc users currently enjoy.

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

ROUNDING NUMBERS - Peter Sandford

Remember how at school you had to 'round' all those decimal numbers? Basic doesn't provide a function to round them directly but you can print them rounded. The function below will do this for you. For example, to print 6.1234 rounded to two decimal places, you'd use:

```
PRINT FNround(6.1234,2)
6.12
```

```

1000 DEF FNround(number,places)
1010 LOCAL @%
1020 @%=&1020009+256*places
1030 =VAL(STR$(number))

```


POWER BOAT RACE

by Charles Francis



Powerboats is a game in which you, and possibly your partner, have to race down the ubiquitous river El Yako with great skill and courage in the attempt to catch the computer controlled 'drone' boat. If you manage to beat this boat in all six heats then you win the 'Golden Cup' prize.

Written by the author of the superb game 'Flowers of Hell', Powerboats represents a good entertaining game to keep the entire family satisfied over the Christmas break. Although a little long, the program is very efficient, and demonstrates how to use your Electron very well.

Once you have typed the program in and saved it onto tape, you can run it. The program will prompt you as to whether a one or two player game is wanted. You should respond to this question by pressing 'Y' if you want the computer to control boat two or 'N' if you want another person to control boat two. Player one uses the 'Z' and 'X' keys for left and right movement, with the throttle of his boat being controlled by the 'C' key. If a two player game has been selected, then the keys '<' '>' and '?' are used by player two to control his boat.

The course of the river is varied and treacherous, with many large slime covered rocks protruding from the surface to hamper your progress; if your boat strikes one of these rocks, then your boat is slowed down and you may lose your place in the race.

Note that a degree of flexibility has been built into the program. In line 3070, D% specifies the river width, B% the difficulty factor, and A% indicates whether you want sound or no. A%=1 means you do, and A%=0 means you don't. The strange loop between lines 170 and 290 is equivalent to, but faster than, C%=0:REPEAT...UNTIL C%.

```
10 REM PROGRAM POWER BOATS
20 REM VERSION E0.3
30 REM AUTHOR C.FRANCIS
```

```
40 REM ELBUG    DECEMBER 1984
50 REM PROGRAM SUBJECT TO COPYRIGHT
60 :
100 ON ERROR GOTO 3100
110 MODEL
120 DIM S%280,Z%150,O%150,W%150,winne
r$(7),time(7),boatleft$(2),boatright$(2
),boat$(2),aft$(2)
130 PROCchar:PROCstring:PROCenv
140 PROCinstruct
150 FOR round%=1TO6:PROCwidth
160 PROCstart
170 FOR C%=0TO0STEP0
180 SOUND&10,2*A%,4,255
190 I%=I%-2*(I%<150):J%=J%-2*(J%<150)
200 ON POINT(G%+8,H%)GOTO210,210,220
ELSE ON POINT(G%+52,H%)GOTO210,210,220
ELSE230
210 IFQ%-27Q%=Q%+1:H%=H%-32:I%=I%-1
220 SOUND&11,A%,0,255:I%=I%-1
230 ON POINT(E%+8,F%)GOTO240,240,250
ELSE ON POINT(E%+52,F%)GOTO240,240,250
ELSE260
240 IFV%-27V%=V%+1:J%=J%-1:F%=F%-32
250 SOUND&12,A%,0,255:J%=J%-1
260 PROCmove
270 K%=SGNK%:L%=SGNL%:P%=P%+K%:G%=P%*
32:M%=50*(K%+1):U%=U%+L%:E%=U%*32:N%=50
*(L%+1):K%=0:L%=0
280 PROCriver:PRINTTAB(R%,0)$(S%+T%)TA
B(RND(D%)+R%,0)G$CHR$253TAB(X%,2)$(Z%+Y%
)TAB(P%,Q%)$(O%+M%)TAB(U%,V%)$(W%+N%);
290 NEXT
300 PROCendround
310 NEXT
320 PROCendgame
330 IF g$="Y"THEN RUN
340 MODE 6
350 END
360 :
1000 DEFPROCchar
1010 VDU23,1,0;0;0;0;
1020 VDU23,224,1,3,7,14,13,27,30,61
1030 VDU23,225,128,192,224,112,176,216
,120,188
```




```

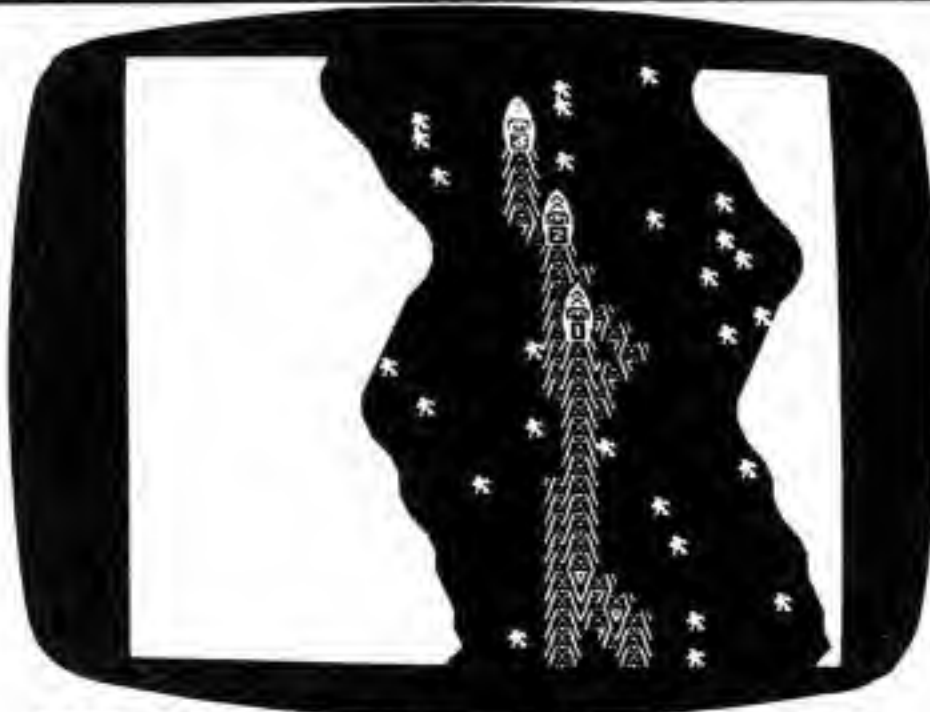
1040 VDU23,226,59,63,48,103,109,105,10
4,127
1050 VDU23,227,220,252,12,230,182,150,
22,254
1060 VDU23,228,120,121,123,57,57,57,56
,63
1070 VDU23,229,30,158,158,156,156,156,
28,252
1080 VDU23,230,112,115,118,49,51,55,48
,63
1090 VDU23,231,14,206,110,204,12,236,1
2,252
1100 VDU23,232,112,119,112,49,54,51,48
,63
1110 VDU23,233,14,238,110,204,108,204,
12,252
1120 VDU23,234,72,74,144,145,16,34,32,
37
1130 VDU23,235,18,82,9,137,8,68,4,164
1140 VDU23,236,255,255,255,255,255,255
,255,255
1150 VDU23,237,255,127,127,63,63,31,31
,15
1160 VDU23,238,15,7,3,1,0,0,0,0
1170 VDU23,239,255,255,255,255,255,127
,63,31
1180 VDU23,240,15,7,7,3,3,1,1,0
1190 VDU23,241,0,1,1,3,3,7,7,15
1200 VDU23,242,31,63,127,255,255,255,2
55,255
1210 VDU23,243,0,0,0,0,1,3,7,15
1220 VDU23,244,15,31,31,63,63,127,127,
255
1230 VDU23,245,0,128,128,192,192,224,2
24,240
1240 VDU23,246,240,248,252,254,255,255
,255,255
1250 VDU23,247,0,0,0,0,0,128,192,224
1260 VDU23,248,240,248,248,252,252,254
,254,255
1270 VDU23,249,255,254,254,252,252,248
,248,240
1280 VDU23,250,224,192,128,0,0,0,0,0
1290 VDU23,251,255,255,255,255,254,252
,248,240
1300 VDU23,252,240,224,224,192,192,128
,128,0
1310 VDU19,0,4,0,0,0:VDU19,1,3,0,0,0:V
DU19,2,2,0,0,0
1320 VDU23,253,112,58,127,249,188,54,3
9,101
1330 ENDPROC

```

```

1340 :
1350 DEFPROCstring
1360 C$=CHR$17:BB$=C$+CHR$128:B$=C$+CH
R$0:GB$=C$+CHR$130:WH$=C$+CHR$3:Y$=C$+C
HR$1:G$=C$+CHR$2
1370 N$=CHR$11:S$=CHR$10:W$=CHR$8:E$=C
HR$9:SWW$=S$+W$+W$:SWWW$=SWW$+W$:SP$=CH
R$32
1380 bow$=CHR$224+CHR$225:mid$=CHR$226
+CHR$227:wash$=CHR$234+CHR$235
1390 aft$(0)=CHR$232+CHR$233:aft$(1)=C
HR$228+CHR$229:aft$(2)=CHR$230+CHR$231
1400 FORi%=0TO2
1410 boatleft$(i%)=bow$+SP$+SWWW$+mid$
+SP$+SWWW$aft$(i%)+SP$+SWWW$+WH$+wash$
+SP$
1420 boatrigh$(i%)=bow$+SWWW$+SP$+mid$
+SWWW$+SP$aft$(i%)+SWWW$+SP$+WH$+wash$
1430 boat$(i%)=bow$+SWW$+mid$+SWW$aft
$(i%)+SWW$+WH$+wash$
1440 NEXT
1450 $Z%=boatleft$(0):$(Z%+50)=boat$(0
):$(Z%+100)=boatright$(0)
1460 FORi%=1TO2
1470 boatleft$(i%)=Y$+boatleft$(i%)
1480 boatrigh$(i%)=Y$+W$+SP$+boatrigh
t$(i%)
1490 boat$(i%)=Y$+boat$(i%)+SWW$+wash$
1500 NEXT
1510 $O%=boatleft$(1):$(O%+50)=boat$(1
):$(O%+100)=boatright$(1)
1520 $W%=boatleft$(2):$(W%+50)=boat$(2
):$(W%+100)=boatright$(2)
1530 ENDPROC
1540 :
1550 DEFPROCinit
1560 E%=D% DIV2:R%=19-E%:X%=19
1570 P%=16:Q%=27:U%=22:V%=27:T%=120
1580 I%=0:J%=0
1590 Y%=50:M%=50:N%=50
1600 E%=U%*32:F%=(32-V%)*32:G%=P%*32:H
%=(32-Q%)*32
1610 ENDPROC
1620 :
1630 DEFPROCstring2
1640 $$%=CHR$238+CHR$239+STRING$(D%-1
),CHR$236)+CHR$246+CHR$247
1650 $(S%+40)=CHR$237+STRING$(D%,CHR$2
36)+CHR$245
1660 $(S%+80)=CHR$240+STRING$(D%,CHR$2
36)+CHR$248

```

```

1670 $(S%+120)=STRING$(D%+1,CHR$236)
1680 $(S%+160)=CHR$244+STRING$(D%,CHR$
236)+CHR$252
1690 $(S%+200)=CHR$241+STRING$(D%,CHR$
236)+CHR$249
1700 $(S%+240)=CHR$243+CHR$242+STRING$
(D%-1,CHR$236)+CHR$251+CHR$250
1710 FORC%=S%TOS%+240STEP40:SC%=GB$+N$
+B$+SC%+BB$:NEXT
1720 ENDPROC
1730 :
1740 DEFPROCriver:ON T% DIV40 GOTO1770
,1790,1810,1830,1850,1870 ELSE1750
1750 IF R%=0 T%=200:Y%=50 ELSE IF RND(
2)-1T%=40:Y%=50 ELSE R%=R%-1:X%=X%-1:Y%
=0
1760 GOTO1880
1770 IF RND(2)=1 T%=120:Y%=50 ELSE T%=
80:R%=R%-1:X%=X%-1:Y%=0
1780 GOTO1880
1790 IF R%=0 T%=200:Y%=50 ELSE IF RND(
2)=1 T%=40:Y%=50 ELSE T%=0:R%=R%-1:Y%=0
:X%=X%-1
1800 GOTO1880
1810 IF RND(2)=1 T%=160:Y%=50:ELSE T%=
80:R%=R%-1:X%=X%-1:Y%=0
1820 GOTO1880
1830 IF R%+D%=38T%=40:Y%=50 ELSE IF RN
D(2)=1 T%=200:Y%=50 ELSE T%=240
1840 GOTO1880
1850 R%=R%+1:X%=X%+1:Y%=100:IF RND(2)=
1 T%=120 ELSE T%=160
1860 GOTO1880
1870 R%=R%+1:X%=X%+1:Y%=100:IF R%+D%=3
8T%=40 ELSE IF RND(2)=1T%=200 ELSE T%=2
40
1880 ENDPROC
1890 :
1900 DEFPROC1L
1910 ON POINT(G%-24,H%-32)GOTO1920,193
0,1950 ELSE ON POINT(G%-24,H%-96)GOTO19
20,1930,1950 ELSE1960
1920 PROC2L
1930 SOUND&11,3*A%,240,10:IF Q%-27Q%=Q
%+1:H%=H%-32
1940 ENDPROC
1950 SOUND&11,A%,0,255:I%=I%-1
1960 I%=I%-1:K%=K%-1
1970 ENDPROC
1980 :
1990 DEFPROC2L
2000 ON POINT(E%-24,F%-32)GOTO2010,202
0,2040 ELSE ON POINT(E%-24,F%-96)GOTO20
10,2020,2040 ELSE2050
2010 PROC1L
2020 SOUND&12,3*A%,240,10:IF V%-27V%=V
%+1:F%=F%-32
2030 ENDPROC
2040 SOUND&12,A%,0,255:J%=J%-1
2050 J%=J%-1:L%=L%-1
2060 ENDPROC
2070 :
2080 DEFPROC1R
2090 ON POINT(G%+84,H%-32)GOTO2100,211
0,2130 ELSE ON POINT(G%+84,H%-96)GOTO21
00,2110,2130 ELSE2140
2100 PROC2R
2110 SOUND&11,3*A%,240,10:IF Q%-27Q%=Q
%+1:H%=H%-32
2120 ENDPROC
2130 SOUND&11,A%,0,255:I%=I%-1
2140 I%=I%-1:K%=K%+1
2150 ENDPROC
2160 :
2170 DEFPROC2R
2180 ON POINT(E%+84,F%-32)GOTO2190,220
0,2220 ELSE ON POINT(E%+84,F%-96)GOTO21
90,2200,2220 ELSE2230
2190 PROC1R
2200 SOUND&12,3*A%,240,10:IF V%-27V%=V
%+1:F%=F%-32
2210 ENDPROC
2220 SOUND&12,A%,0,255:J%=J%-1
2230 J%=J%-1:L%=L%+1
2240 ENDPROC
2250 :
2260 DEFPROCUI1:IF I%<0 ENDPROC
2270 SOUND&11,A%,10,255:H%=H%+32:Q%=Q%
-1:I%=I%-B%:IF Q%=0C%=1
2280 ENDPROC
2290 :
2300 DEFPROCUI2:IF J%<0 ENDPROC
2310 SOUND&12,A%,10,255:F%=F%+32:V%=V%
-1:J%=J%-B%:IF V%=0C%=2
2320 ENDPROC
2330 :
2340 DEFPROCenv
2350 ENVELOPE1,1,4,-2,-2,2,2,2,126,0,0
,-126,126,126
2360 ENVELOPE2,1,-1,0,0,2,0,0,126,0,0,
-126,126,126
2370 ENVELOPE3,1,-12,-8,-2,5,10,40,126
,0,0,-126,126,126
2380 ENDPROC
2390 :
2400 DEFPROCinstruct
2410 won1%=0:won2%=0

```




```

2420 CLS
2430 PRINTTAB(14,2)Y$"POWER BOATS"
2440 PRINTTAB(9,4)G$"A game for two pl
ayers"
2450 PRINTTAB(5,6)"Do you want me to c
ontrol boat 2?"
2460 REPEAT
2470 g$=GET$
2480 UNTIL INSTR("YN",g$)>0
2490 IF g$="Y"THEN e%=TRUE ELSE e%=FAL
SE
2500 COLOUR1
2510 INPUTTAB(0,8)"What is your name,
player 1? "name1$
2520 IF e% name2$="ELK" ELSE INPUTTAB(
0,10)"What is your name, player 2? "nam
e2$
2530 ENDPROC
2540 :
2550 DEFPROCstart
2560 PROCinit:PROCstring2
2570 FOR C%=0 TO15:PRINTTAB(R%,0)$(S%+
T%);:NEXT
2580 FOR C%=0 TO15:PROCriver
2590 PRINTTAB(R%,0)$(S%+T%)TAB(RND(D%
+R%,0)G$CHR$253
2600 NEXT
2610 PRINTTAB(X%,2)$(Z%+50)TAB(P%,Q%)$
(O%+50)TAB(U%,V%)$(W%+50);
2620 PRINTTAB(10,30)"Press SPACE to st
art"
2630 REPEAT UNTIL INKEY-99
2640 TIME=0
2650 ENDPROC
2660 :
2670 DEFPROCendround
2680 CLS
2690 SOUND3,0,0,1
2700 time%=TIME:time(round%)=time% DIV
6000+(time% MOD6000)/10000
2710 IF C%=1 THEN winner$(round%)=name
1$:won1%=won1%+1:ELSE winner$(round%)=n
ame2$:won2%=won2%+1

```

```

2720 COLOUR3
2730 PRINTTAB(16,1)"ROUND ";round%
2740 @%=&20205
2750 COLOUR1:PRINTTAB(2,5)winner$(roun
d%) " won in ";time(round%);" minutes"
2760 PRINTTAB(11,10)WH$"VICTORY TABLE"
TAB(2,10)"ROUND PLAYER"SPC(16)"TIME"
2770 FOR c%=1 TO round%
2780 @%=&0090A
2790 PRINTTAB(4,11+2*c%);c%
2800 @%=&20205
2810 PRINTTAB(9,11+2*c%)winner$(c%)TAB
(30,11+2*c%)time(c%)
2820 NEXT
2830 @%=&0090A
2840 IF round%-7 PRINTTAB(9,30)"Press S
PACE to continue":REPEAT UNTIL INKEY-99
2850 ENDPROC
2860 :
2870 DEFPROCendgame
2880 IF won1%>won2% PRINTTAB(1,28)name
1$;" wins ";won1%;" rounds to ";won2%;:
GOTO2900
2890 PRINTTAB(1,28)name2$;" wins ";won
2%;" rounds to ";won1%;
2900 PRINTTAB(1,30)"Do you want anothe
r game?";SPC(7);
2910 REPEAT:g$=GET$:UNTIL INSTR("YN",g
$)
2920 ENDPROC
2930 :
2940 DEFPROCmove
2950 IF INKEY-83 PROCU1:GOTO2980
2960 IF INKEY-98 PROC1L
2970 IF INKEY-67 PROC1R
2980 IF e%THEN3030
2990 IF INKEY-105 PROCU2:GOTO3020
3000 IF INKEY-103 PROC2L
3010 IF INKEY-104 PROC2R
3020 ENDPROC
3030 IF POINT(E%+8,F%+32)=2 OR POINT(E
%+8,F%)=2 OR POINT(E%+8,F%+32)=1 PRINTT
AB(U%,V%-2)SP$SP$;:PROC2R ELSE IF POINT
(E%+52,F%+32)=2 OR POINT(E%+52,F%)=2 OR
POINT(E%+52,F%+32)=1 PRINTTAB(U%,V%-2)
SP$SP$;:PROC2L ELSE PROCU2
3040 ENDPROC
3050 :
3060 DEFPROCwidth
3070 D%=20:B%=20:A%=1
3080 ENDPROC
3090 :
3100 ON ERROR OFF:MODE 6
3110 IF ERR=17 END
3120 REPORT:PRINT" at line ";ERL:END

```


ALLIGATA'S CONTRACT BRIDGE

Reviewed by Sheridan Williams

Program : Contract Bridge
Supplier : Alligata Software Limited
Price : £9.95

As far as I am aware this is the only program available on the Electron/BBC micro which will bid and play the other three hands in a game of Bridge.

THE DEAL

There is obviously an algorithm in the random deal to produce "interesting" hands. There is no other explanation for the number of voids, six and seven card suits that are dealt. The more hands I played the more infuriating this became; part of the fun of Bridge is playing the odds; if you know these have been fiddled you cannot play in a calculated way.

THE BIDDING

The computer randomly decides who opens the bidding, but you are not given any details of vulnerability. The package says the bidding is natural and has "Stayman", pre-emptive jump bids, and strong two bids.

I got very confused by some of the bids, which seemed most unnatural. I can only liken it to bidding with someone in whom you have little confidence. One quite common bid is "STOP 3 SPADES" whatever this means. A disappointing omission from the bidding was "doubling", and this essential means of communication could not be used. Nevertheless, better to have a novice partner than not being able to play at all, and after several hours play I began to adapt to the handicap.

THE PLAY

The computer as your partner cannot be trusted to notice your signals, e.g. "high low" play with a doubleton. With the computer playing your opponents'



hands, it seems uncannily able to play the right card at critical moments. The package says that it plays a rubber with scoring, but I could not make it do this. Each hand is totally separate and does not go to make up a rubber.

THE PROGRAM

When examining the program it is obvious why some features or enhancements are missing, this is because the program uses virtually all the memory already. Compressing the program (I used BEEBUGSOFT's TOOLKIT) saves over 1700 bytes and would allow considerable enhancements to be incorporated. There is also a 'subscript out of range' bug which manifests itself in a long bidding sequence. This does not occur often but it is a pity it has not been cured.

CONCLUSION

I would not profess to be an expert Bridge player, but I have played on and off for 20 years. I found this program to be worth buying simply because it provides an opportunity to play Bridge when you cannot make up a four.

I learned to live with its shortcomings and until a better version is sent for review I can recommend Alligata's Contract Bridge.



CHRISTMAS CAROL

by D. G. Chappell

With Christmas approaching we have decided to include a program with a seasonal flavour that provides a good demonstration of the sound and graphics capabilities of your Electron. The program provides an attractive Christmas card display on the screen while playing a rendering of "See Amid The Winter's Snow". We are sure that you will find the results of typing in this relatively short program well worth while. Be careful when copying the program, particularly with the data statements at the end, otherwise the music will be out of tune.

The program is well structured and you should have no difficulty in identifying the various parts. The music is held as a series of numbers in data statements which are used as the parameters for sound, frequency and duration in the procedure PROCmusic.

Once started the music will continue indefinitely until you press Escape or Break.

```

10 REM PROGRAM XMAS CARD
20 REM VERSION E0.1
30 REM AUTHOR D.G.CHAPPELL
40 REM ELBUG DECEMBER 1984
50 REM PROGRAM SUBJECT TO COPYRIGHT
60 :
100 ON ERROR GOTO 1160
110 MODE2
120 GCOL0,129:CLG
130 VDU5:GCOL0,2
140 MOVE120,600
150 PRINT"MERRY CHRISTMAS"
160 GCOL0,7:MOVE128,604
170 PRINT"MERRY CHRISTMAS"
180 W%=INKEY(500)
190 MODE4:VDU19,0,4,0,0,0
200 VDU23,1,0;0;0;0;
210 DIM S(7),C(7)
220 CLS:A=-PI/3
230 W%=INKEY(100):RESTORE
240 FORI%=0TO7:S(I%)=SIN(A)
250 C(I%)=COS(A):A=A+PI/3
260 NEXT
270 X%=350:Y%=750
280 :
290 REPEAT:VDU29,X%;Y%;

```

```

300 A%=250:B=RND(1)*0.4+0.3
310 FORI%=0TO240 STEP15
320 FORK%=1TO6
330 MOVE0,0:MOVEI%*S(K%),I%*C(K%)
340 PLOT1,A%*S(K%-1),A%*C(K%-1)
350 PLOT1,-A%*S(K%-1),-A%*C(K%-1)
360 PLOT1,A%*S(K%+1),A%*C(K%+1)
370 NEXT
380 PROCmusic:A%=RND(240-I%)*B
390 NEXT
400 X%=X%+500+RND(100)
410 IFX%>1000THENX%=300+RND(100)
420 FORI%=0TO7
430 W%=INKEY(15):PRINTTAB(0,0);:VDU11
440 NEXT
450 UNTILFALSE
460 END
470 :
1000 DEFPROCmusic
1010 IEADVAL(-6)<3 THEN ENDPROC
1020 READP%,D%
1030 IFD%=1THENRESTORE:ENDPROC
1040 IFD%<2THEN SOUND1,0,0,(1+30*D%)
1050 SOUND1,-15,P%,D%
1060 ENDPROC
1070 :
1080 REM ** SEE AMID THE WINTERS SNOW
1090 DATA129,18,137,6,129,12,125,12,11
7,12,117,6,109,6,109,12,109,12
1100 DATA129,12,137,12,149,12,145,12,1
45,12,145,6,137,6,137,24
1110 DATA129,12,129,6,137,6,129,12,125
,12,117,12,117,6,109,6,109,12,109,12
1120 DATA129,12,137,12,145,12,149,12,1
37,12,137,6,129,6,129,24,157,17,0,0
1130 DATA157,6,149,12,145,12,137,12,12
9,12,125,24,157,17,0,0,157,6,149,12,145
,12,137,12,129,12,125,24
1140 DATA129,18,137,6,129,12,125,12,11
7,12,117,6,109,6,109,12,109,12,157,18,1
45,6,129,12,149,12,145,12,137,12,129,24
,0,1
1150 :
1160 ON ERROR OFF:MODE 6
1170 IF ERR=17 END
1180 REPORT:PRINT" at line ";ERL
1190 END

```




Title : Elite
 Supplier : Acornsoft
 Price : £12.95
 Reviewer : David Fell
 Rating : *****

Acornsoft Elite is the first in a new generation of 3-D space games featuring interstellar travel in a distant cluster of galaxies, or so goes the advertising - unusually, it's right. I've played Elite for a week now and I'm convinced that it is the best game ever for the Electron.

Elite combines the elements of a number of classical games to produce a superb three dimensional graphical trading game of skill and absolute addiction, that squeezes every ounce of performance out of the Electron. Elite combines a three dimensional action combat game, a 'Monopoly' like trading game and various aspects of an adventure style exploration game.

Elite takes its name from the overall goal of the game, which is to achieve a rating of 'Elite'. To achieve this you need to be a successful space trader (to purchase the necessary weapons) and achieve a high level of flying and combat skills (your rating is based upon the latter, and your survival is dependent upon both).

Your space ship is equipped with normal and hyperspatial drives. Combat, normal flight and docking are all performed in real time, with some

stunning graphics achieving an amazing impression of realism.

The trading is quite complex due to the sheer quantity of information involved: 2000 planets selling 17 different items; political situation; planetary produce, etc. If all this sounds too dull you can always give up the life of the honest trader and become a bounty hunter or space pirate, but this involves living life dangerously.

Combat takes a while to master, and requires sheer determination and three dimensional perception to survive. Your armoury, apart from three missiles and a small laser, will depend upon accruing money from trading. As a warning, it is said that there are spaceships 'out there' that no one has ever seen!



Elite is undoubtedly a masterpiece of programming and I would recommend anyone who has an Electron to buy a copy as soon as possible. Elite has already firmly established itself as a cult game for the BBC micro and looks set to do the same for the Electron.

Title : Sadim Castle
 Supplier : MP Software
 Price : £7.50
 Reviewer : Mitch
 Rating : ***

Having been foully murdered by her husband, the Lady Leonara has cursed the castle and all who set foot within it. It was my task to find her mortal remains and lay them to rest in peace.

You can also see:
The gate is closed

You are on a track which runs east & south beside a high wall, in the north green fields & a wooden gate can be seen

You are on a track that runs N/S beside a high wall, a path branches west & a sign reads TO THE HOLY SPRING

You are at the end of a rocky valley which stretches east. Water gushes from beneath a boulder & trickles away across the stony ground

Time passes.....

Command> S
Command> S
Command> W
Command> _

The present ghastly inhabitants didn't seem impressed by my chivalry, however, and did their damndest to kill me. A few kindly souls did assist me, if I first solved their problems, so it does help to be polite! The game is a coloured text, split screen affair with lots of well written descriptions. The castle abounds with locked doors which require an endless supply of keys. Opening doors becomes such a confusing business that I found I spent an hour trying to unlock one which I later found wasn't even locked!

Not so easy is the solution to the wandering monster that follows you. Unfortunately he continues to reappear with monotonous regularity causing you to repeat the same action ad-infinitum. The game however has a nice 'feel' to it and is not too difficult.

Title : The Valley Of The Kings
Supplier : MP Software
Price : £7.50
Reviewer : Mitch
Rating : **

A similar format of coloured text and split screen, this game found me crawling around in decaying corpses beneath the pyramid of old King Tut. The place is so hot and smelly that you'd better ensure you bring lots to drink and something to keep the deadly stink out of your nostrils. I found a laundry basket which hissed and a pit in which something nasty was slithering!

A real time element coupled to your endless thirst and flickering torch has

been added to the game. You have no time to stand and stare but must hurry ever forward towards that light at the

large tent

You are by a group of hobbled camels. Tracks lead north & west

You are wandering around the archaeologists camp. Tracks lead south, west & east towards a large tent

You are in a long arid valley at the middle of an archaeologists camp. A track leads away north

You are by a group of hobbled camels. Tracks lead north & west

You are thirsty
A bad tempered camel spits at you
Time passes.....

Command> H
Command> M
Command> E
Command> _

end of the tunnel, which knowing my luck will be a train coming the other way!

This game has all the usual ingredients, but in that is its weakness. I couldn't find that magic something which is needed to keep me battling forward so I folded my tent, mounted my camel and stole off home.

Title : Gauntlet.
Supplier : Micro Power.
Price : £6.95
Reviewer : Geoff Bains.
Rating : ****



Gauntlet shows just what can be done with an Electron if you try. Gauntlet is a version of the now ageing arcade classic, Defender, in which you have to defend a planet's surface, and more importantly the beings upon it, from the invaders. The invaders come in all

forms. The most important are those that actually kidnap the planet's natural inhabitants. You can perform thrilling mid air rescues to return the beings to the surface. Other invaders are just there to get you. Some split up when hit to form several very fast moving smaller critters.

To aid you in this quest you have but a small craft equipped with a long range scanner and a front-facing laser and a few 'smart bombs' which destroy every hostile thing visible on the screen.

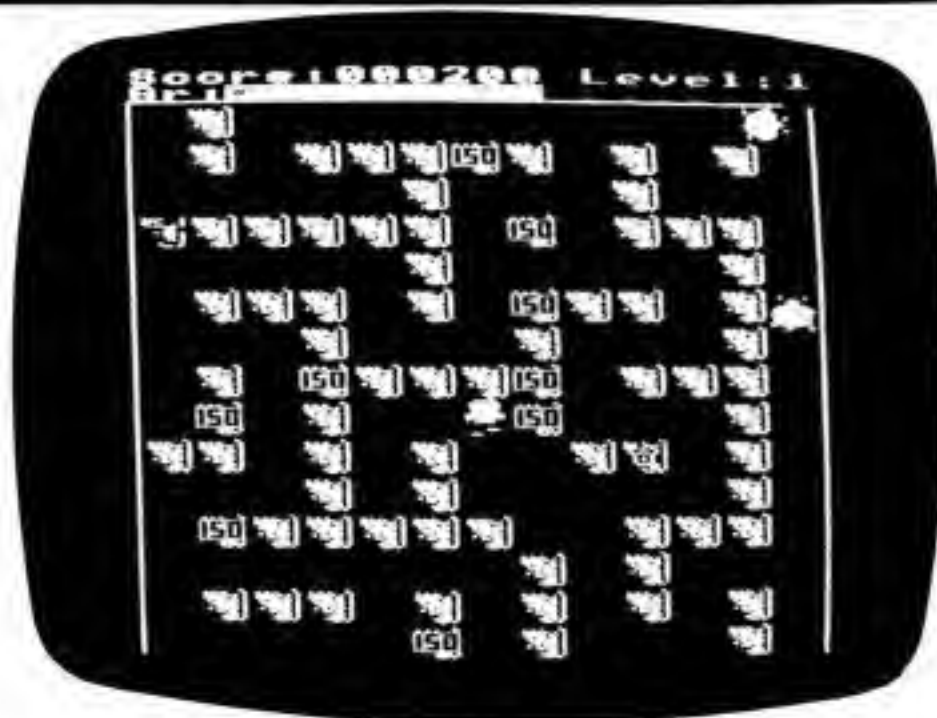
As such, Gauntlet is a fairly standard version of Defender. That is what is so good about it. Not so long ago nobody thought it possible to perform such smooth moving, action-filled games as this on an Electron. Micro Power has proved the point very convincingly. The graphics are excellent. All the action is very fast and smooth, the controls responsive, and the colours well thought out. There is so much going on all the time. Some of the 'set pieces' such as your craft exploding (it does happen occasionally!) are brilliant - better than the original or Acornsoft's much acclaimed BBC version.

As if all this wasn't enough there is even a demonstration mode. If your fingers get worn too close to the bone you can sit back and watch it all happen with the Elk at the controls. Marvellous.

Next to a masterpiece such as Elite I guess Gauntlet still looks pretty tame, but it is an excellent version of an old favourite and as such will probably do as well for Micro Power as it will for you.

Title : Rubble Trouble
Supplier : Micro Power
Price : £6.95
Reviewer : Alan R Webster
Rating : ****

Rubble Trouble is one of a number of new releases from Micro Power which have been converted from the BBC micro. The game is set in a post-nuclear



wasteland and involves crushing mutant 'Krackats' by pushing boulders onto them. The game is similar in concept to the arcade game 'Pengo', except that the scenery is different and when you push a boulder it bounces back and forth across the screen. This can be very dangerous, especially if you happen to be in the way of the rebound.

There are also bonus rocks strewn across the screen which are very useful for points. There are 3 different scenarios to this game ranging from the simple 'Pengo' type screen to one where the whole playing area is full of blocks, and you have to find your way through (this is much more difficult than it sounds).

The graphics are fast and smooth, and the sound effects excellent. There are also comprehensive instructions (including weird sounds) and a high score table. Overall this game is very good, and certainly a novelty. If you've played 'Pengo', then you'll love Rubble Trouble.

Title : Espionage
Supplier : Modular Resources Ltd.
Price : £8.95
Reviewer : Geoff Bains
Rating :

Espionage is (apparently) 'a unique game of intrigue...programmed to assess your intelligence'. Presumably assessment is on the basis of whether you buy the game, or not.

FROM: The Personnel Department.

JOB ALLOCATION

You are employed as
-OUR LAB TECHNICIAN

You are based
- AT A PRODUCTION PLANT

The location is
- SOMEWHERE IN THE CARIBBEAN

Your work is particularly concerned
with - BENCH RIGS

PRESS THE SPACE BAR TO CONTINUE

In Espionage you have to discover the identity of a mole who is leaking information from your Oil Company to a rival firm. To do this you answer boring telexes, memos, and so on from other members of staff, trying to see if they are interested in subjects outside their field.

The whole game is written in badly structured Basic. It has as much interest as a chemistry exam. Even when you do discover the mole, all you get is a curt 'well done' and it's back to square one again.

The worst feature of this game, however, is that it is in several sections, accessed in a different order each game. This means that finding the next section to load takes ages, especially as each section is over 50 blocks long. The messages displayed, such as 'Rewind side two of the tape to just after 170 on the tape counter' are pretty useless as soon as you realize that different recorders have differing tape counters. On the official Acorn data recorder this estimate was about five minutes worth out.

Espionage is certainly a novel game. It's the first of its type. Let's hope the last, too.

Oh yeah, you get a free sticker, pen, and card too. A real bargain at nearly nine quid.

Name : Electron Zalaga
Supplier : Aardvark Software
Price : £7.90
Reviewer : David Fell
Rating : ****



Electron Zalaga is a new game from the writer of classics such as Frak! (for the Beeb), and Arcadians (for both the Beeb and the Electron). The game exemplifies what can be achieved on the Electron [see also the 'Elite' review - Ed.], and provides a challenging and enjoyable fast action combination of skill and luck.

Based upon the arcade game Galaga, Zalaga is a good example of the zap 'em quick type game. After an attractive mode 4 title page, all the action takes place in mode 5, with three 'layers' of different colour stars providing a continual moving backdrop, even whilst entering your high score! Aliens swoop in from the top and sides of the screen, and assemble themselves into various patterns, prior to near suicidal attacks upon yourself. You are armed with a powerful laser base, and can use this very effectively to cut through the hordes hovering above you. Options for auto-repeat and speed of reload for your laser gun are included via a special 'Operator mode'. This mode also caters for a choice of keyboard or joystick (analogue or digital) control, and the selection of a one or two player game.

Electron Zalaga is a good game and should be considered as a possibility for all game players' shopping (or Christmas!) lists.

THREE SHORT GAMES

by N. Silver

Although most of the games that appear in Elbug stretch over many pages, it is possible to produce an entertaining game in just a few lines. Here we present three games that you can type into your Electron between the turkey and the plum pudding on Christmas day.

Each of these three games is a masterpiece of miniaturization in its own right. Each can, in fact, be crammed into a single program line though here they are presented in a slightly more readable form. Although they each take but a few seconds to type in, they will keep you happy for hours.

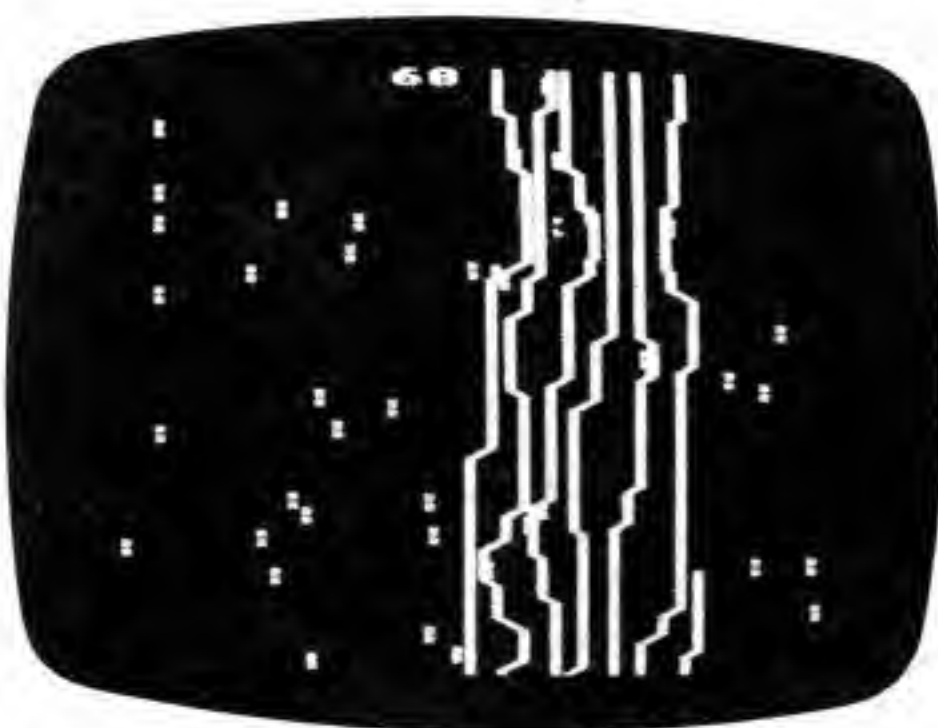


TREASURE HUNT

Treasure hunt is a game of pure luck combined with a touch of skill. The display is simple, as are the rules. You control the asterisk in the centre of the screen with the usual keys: Z and X for left and right and * and ? to move up and down. At the top of the screen there are displayed two figures. The left hand one is the number of moves that you have made so far. The right hand figure is the distance, in character positions, to the hidden treasure. You have to guess the direction to the treasure, with the help of the distance readout, and get there in the minimum of moves.

Once you're there, the game finishes. You can start a new game by pressing the space bar.

```
10 @%=778:X=20:Y=11:S=0:D=0
20 A=RND(40)-1:B=RND(11)
30 MODE 6:VDU 23,1,0;0;0;0;
40 REPEAT
50 C=INKEY 9
60 X=X+(C=90)-(C=88):X=(X+40)MOD40:Y
=Y+(C=58)-(C=47):Y=(Y+22)MOD23+1
70 S=S+SQNINSTR("ZX/:",CHR$C):D=SQR(
ABS(A-X)^2+ABS(B-Y)^2)
80 VDU 7807;
90 PRINTS,D
100 VDU 31,X,Y,42
110 UNTIL D=0:VDU 7
120 REPEAT UNTIL GET=32:RUN
```



TRUFFLE HUNT

Truffle hunt is a game in colour, in only 16 short lines of Basic. In case you have never seen one, truffles (for this game, at any rate) look remarkably like yellow colons! As you drift ever upwards you have to steer around the screen (field), using the Z and X keys for left and right, colliding with as many truffles as you can. All the time you leave a white trail behind you. You must be careful to avoid this trail.

You must also avoid the red asterisks (poisonous mushrooms) that litter the field. As soon as you disappear off the top of the screen,

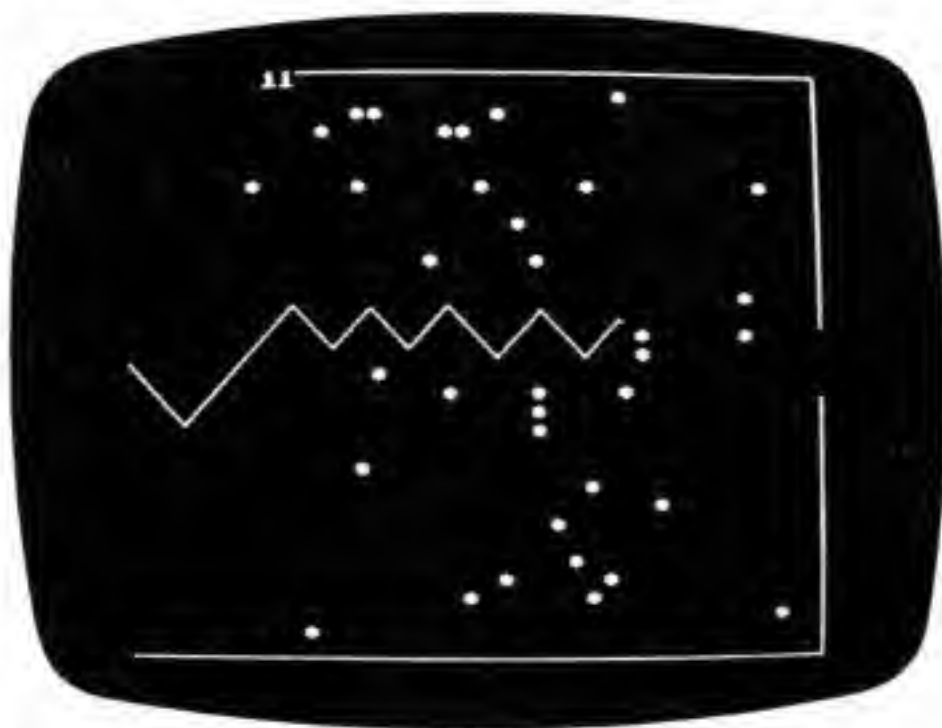
you re-appear at the bottom and as you drift off the side you'll appear again on the opposite edge of the screen.

Truffle Hunt is a fast game so keep your wits about you. The space bar will restart the game if (when!) you munch a mushroom.

```

10 S%=0:X%=640:Y%=0
20 MODE5
30 FOR I%=1 TO 50
40 VDU 5,18;1,25,4,RND(1270);RND(102
3);42,18;2,9,9,124,18;3
50 NEXT I%
60 REPEAT
70 PLOT 69,X%,Y%
80 DRAWX%+8,Y%:X%=X%+8*(INKEY-98-INK
EY-67)
90 X%=(X%+1280)MOD 1280
100 Y%=Y%+4:Y%=Y% MOD 1024
110 P%=POINT(X%,Y%):S%=S%-(P%=2)
120 UNTIL(1 AND P%)=1
130 VDU 7,4
140 PRINT S%
150 REPEAT UNTIL INKEY-99:RUN
160 END

```



ASTERISK TRACKER

Asterisk Tracker is arguably the best of this suite of mini games. The concept is very simple. The playing area covers the whole of the screen. This is bounded with a white line. At the right hand edge of the screen there

is a small break in this line. Dotted around the screen there are a number of asterisks.

Enter you, stage left. You control a fast moving line that has (as Newton would be pleased to explain) a tendency to descend to the bottom of the screen. Your one control is the Return key which temporarily reverses gravity.

The idea is to steer the advancing line between the asterisks and through the hole in the border. If you once touch the border, or an asterisk, you die!

Managing this the first time is easy. Next time, however, the screen has more asterisks, and the third time even more, and so on, until you fail! The number of the screen that you are playing is displayed at the top of the screen. You can start a new game by pressing the space bar.

```

10 L=0
20 REPEAT
30 MODE 4:L=L+3
40 DRAW 1279,0:DRAW 1279,452
50 MOVE 1279,572:DRAW 1279,1023:DRAW
0,1023
60 FOR I%=1 TO L
70 VDU 31,RND(32)+5,RND(31),42,30
80 NEXT I%
90 PRINT(L-3)/3
100 X=0:Y=512
110 REPEAT
120 PLOT 69,X,Y
130 X=X+4:Y=Y-(INKEY-74+.5)*8
140 UNTIL POINT(X,Y)=1 OR X=1280
150 UNTIL X<1280:VDU 7
160 REPEAT UNTIL INKEY-99:RUN
170 END

```

By the time you've read all this you could have typed in all three games and played four rounds on each of them. What are you waiting for? Start typing and have fun!

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

MACROS IN ASSEMBLER

If you are using functions to generate macros in the Electron's 6502 assembler, then you will find it more reliable to use EQU\$ to call your code and to return a null string than to use OPT and return the current 'pass'. The reason for this is that there are some obscure bugs in the OPT section of Basic which don't occur when using EQU\$.

TEXT TO ORDER

by Alan Dickinson

Alan Dickinson shows how to produce some scintillating displays of text in mode 5, one of the most useful graphics modes on the Electron, yet often rejected for its otherwise poor characters.

Mode 5 is a screen mode which offers four colours, graphics, chunky characters, and uses only 10k of memory. Surely this is an ideal mode for many applications requiring a mixture of text and graphics, such as educational programs aimed at children, and programs which cannot afford the luxury of the 20k of RAM gobbled up by mode 1. Yet it rarely seems to be used, and the reason isn't difficult to see - mode 5 text looks terrible!

PROPORTIONAL SPACING

One way to improve the appearance of text in mode 5 is to design a completely new character set, seven pixels wide, and pack the characters slightly closer together. This requires rather a lot of brain-power, not to mention RAM for the exploded character set. An alternative approach is to use the existing character set, and to proportionally space the text according to the width of each individual character. This form of spacing is the type of spacing we normally see in books, magazines, and so on, and is much more pleasing to the eye than text arranged into strict columns.

Proportional spacing used in mode 5 often allows around 22 characters per line, because characters such as spaces, full stops and commas do not take up the full character cell width. The disadvantage is that all output has to be written via the graphics cursor, and is therefore considerably slower and less convenient for the programmer.

The routine PROCwidth is the key routine for calculating the width required. It uses an Operating System routine called OSWORD to read the character definition, and logically 'OR's together the eight rows of the definition into a single amalgamated row, from which it can deduce the overall width of the character. The amount of redundant space to the left

PROPORTIONAL SPACING

Proportional spacing can fit more characters onto a MODE 5 screen, and it's neater as well.

For example....

123456789.9876543210
123456789.9876543210123

ELBUG Magazine

Christmas 1984

and right of the character is stored in L% and R% for use by the printing routine. In order to make conversion to other modes simpler, W% contains the pixel width for the mode in use. If you use this program in another mode, change W% as follows:

MODE 0 - W%=2
MODE 1 - W%=4
MODE 2 - W%=8
MODE 4 - W%=4
MODE 5 - W%=8

The technique cannot be applied to modes 3 or 6 as these are not graphics modes, an essential requirement.

PROCp is the printing routine which is passed a character size, (1 or 2 only), and the string to be printed. The text is printed at the graphics cursor and in the current graphics foreground colour. If you do not require any double-height text, then all references to the size parameter, s%, and PROCdouble may be eliminated.

BACKLIGHTING OF TEXT

3D lettering effects may be obtained by printing text in black, and then overprinting the text with a slight x and y offset in a lighter colour. The portions of black which protrude from behind the text create a shadow effect. This technique can be



taken further to produce a complete black outline to the letters you are printing, and the WIDTH routine is used to ensure that characters do not overlap. A routine to perform this backlighting is included, PROCb. Again, a size parameter is passed, which can be eliminated if double-height text is not required.

The program includes a couple of demonstration routines, the first illustrates the use of proportionally spaced text, and the second displays the character set as back-lit double-height characters. Adapt these routines to your own requirements, and make your mode 5 programs more attractive.

[ELBUG readers might also like to refer back to previous issues of ELBUG, notably Vol.1 Nos.1,2 and 3, for other ideas on interesting ways of displaying text. Ed.]

```

10 REM Program PROTXT
20 REM Version El.0
30 REM Author Alan Dickinson
40 REM ELBUG December 1984
50 REM Program subject to copyright
60 :
100 MODE5
110 ON ERROR GOTO 600
120 VDU19,2,6,0,0,0
130 VDU5:W%=8
140 GCOLOR,129
150 REM Demonstration 1
160 REM =====
170 :
180 CLG
190 MOVE80,960
200 PROCb(2,"PROPORTIONAL SPACING")

```

```

210 GCOLOR,3
220 MOVE15,800
230 PROCp(1,"Proportional spacing can
")
240 MOVE15,756
250 PROCp(1,"fit more characters onto
")
260 MOVE15,708
270 PROCp(1,"a MODE 5 screen, and it'
s")
280 MOVE15,660
290 PROCp(1,"neater as well.")
300 GCOLOR,2
310 MOVE15,480
320 PROCp(2,"For example....")
330 GCOLOR,3
340 MOVE15,340
350 PRINT"123456789.9876543210"
360 MOVE15,272
370 PROCp(1,"123456789.9876543210123")
380 MOVE15,160
390 PROCb(2,"ELBUG Magazine")
400 MOVE15,60
410 PROCb(1,"Christmas 1984")
420 A$=GET$
430 :
440 REM Demonstration 2
450 REM =====
460 :
470 CLG
480 MOVE340,960
490 PROCb(1,"Backlighting")
500 x%=0:y%=840
510 FOR n%=32 TO 126
520 x%=x%+96
530 IF x%>1160 THEN x%=96:y%=y%-96
540 MOVE x%,y%
550 PROCb(2,CHR$(n%))
560 NEXT
570 A$=GET$:MODE 6
580 END
590 :
600 ON ERROR OFF:MODE 6
610 IF ERR<> 17 THEN REPORT:PRINT" at
line ";ERL
620 END
630 :
1000 REM Proportional spacing
1010 REM =====
1020 :
1030 DEF PROCp(s%,a$)
1040 LOCAL a%,c%
1050 FOR a%=1 TO LEN(a$)
1060 c%=ASC(MID$(a$,a%,1))
1070 PROCwidth(c%)
1080 IF s%=2 THEN PROCdouble
1090 PLOT0,-L%,0
1100 IF s%=1 THEN VDU c%
1110 IF s%=2 THEN VDU255,10,8,254,11
1120 PLOT0,-R%,0

```



```

1130 NEXT
1140 ENDPROC
1150 :
1160 REM Back-lighting
1170 REM =====
1180 :
1190 DEF PROCb(s%,a$)
1200 LOCAL j%,k%
1210 FOR a%=1 TO LEN(a$)
1220 c%=ASC(MID$(a$,a%,1))
1230 PROCwidth(c%)
1240 IF s%=2 THEN PROCdouble
1250 GCOL0,0
1260 PLOT0,-L%-W%,0
1270 FOR j%=-W% TO W% STEP W%
1280 PLOT0,0,-4
1290 FOR k%=-4 TO 4 STEP 4
1300 IF s%=1 THEN VDU c%,8
1310 IF s%=2 THEN VDU255,10,8,254,11,8
1320 PLOT0,0,4
1330 NEXT
1340 PLOT0,W%,-8
1350 NEXT
1360 PLOT0,-2*W%,0
1370 GCOL0,2
1380 IF s%=1 THEN VDU c%
1390 IF s%=2 THEN VDU 255,10,8,254,11
1400 PLOT0,-R%,0
1410 NEXT
1420 ENDPROC
1430 :
1440 REM PROCwidth calculates the size
1450 REM of the blank column on each
1460 REM side of the character passed.
1470 REM Returns values in L% and R%.
1480 REM Assumes pixel-width in W%.
1490 :
1500 DEF PROCwidth(c%)
1510 LOCAL X%,Y%,A%,t%,u%
1520 X%=&70:REM Pointer to
1530 Y%=0:REM data block
1540 A%=&A
1550 ?&70=c%
1560 CALL &FFF1
1570 t%=0
1580 FOR u%=&71 TO &78
1590 t%=t% OR ?u%
1600 NEXT
1610 IF t%=0 THEN L%=0:R%=4*W%:ENDPROC
1620 L%=-1:u%=t%
1630 REPEAT
1640 L%=L%+1:u%=u%*2
1650 UNTILu%>127
1660 R%=-1:u%=t%*2
1670 REPEAT
1680 R%=R%+1:u%=u%DIV2
1690 UNTIL(u%MOD2)=1
1700 L%=L%*W%:R%=R%*W%
1710 ENDPROC
1720 :
1730 REM This procedure defines two
1740 REM characters, 255 and 254, as
1750 REM a double-height version of
1760 REM the last character processed
1770 REM by PROCwidth.
1780 :
1790 DEF PROCdouble
1800 VDU23,255,?&71,?&71,?&72,?&72,?&7
3,?&73,?&74,?&74
1810 VDU23,254,?&75,?&75,?&76,?&76,?&7
7,?&77,?&78,?&78
1820 ENDPROC

```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

EVEN BETTER INPUT?

If you are using PRINT to print out a prompt just before inputting some data, and are finding that you get a '?' each time the Electron waits for data when you don't want it, then use INPUT ""....

NOISE FOR FIREWORKS - M. Scott

If you liked the fireworks program published in ELBUG Vol.1 No.10 then you might like to add the following line for some sound effects whilst the program is running.

```
1015 IF RND(10)<4 SOUND 0,-15,4,1
```

TIDY FORMATTING

If you are writing a program in which you print a lot of numbers, but are finding that they either don't line up, or are taking up too much space, then try setting the special variable @% (see page 60 onwards in the User Guide). The short program below illustrates briefly how this can be used:

```

10 @%=4
20 FOR I=0 TO 10
30 PRINT I,;
40 NEXT I

```


White Knight

Reviewed by Andy Jenkins

Chess player, Andy Jenkins pits his wits against the combined forces of White Knight and the Electron.

Title : White Knight (Mk.II)
 Supplier : BBC SOFT
 Price : £9.95 inc. VAT
 Rating : ****

There are several chess programs available for the Electron but let the buyer beware! The standard varies widely. For a short time it will boost your ego to beat a weak program, but you'll soon tire of that. The latest program from BBC Soft, White Knight, claims to be so strong that even above-average players have to treat it with respect.

Those who find the program too strong can always start off by letting it play itself and pick up some tips that way. White Knight will also solve problems of the mate in 2 or 3 variety.

White Knight has all the standard features and a few novel ones. These are well described in the accompanying 12 page booklet. The display is very clear and uses a two colour graphics mode in black and cyan. The time taken by each 'player' is displayed at the top of the board diagram.

There are three main playing modes: 'Equality', 'Average' and 'Tournament'. the longer White Knight has to think, the stronger it will play. The three modes are just different ways of defining the machine's thinking time. In 'Equality' the machine will roughly keep pace with you. 'Average' requires you to specify an average response time. At the default value of 10 seconds, White Knight will undertake to play 60 moves in 10 minutes but may take longer than 10 seconds over individual moves. 'Tournament' allows you to specify tournament-style time limits if you are really serious e.g. forty moves in two and a half hours, as used in the World Championship.

In all cases you are in control of how quickly White Knight plays - a



great improvement over some rival programs.

While White Knight is thinking it displays the best line it has found, updating this from time to time when it finds better lines. When making a move it gives the number of positions that it has examined and this can vary from about 250 to 250,000 or more when it has been thinking hard.

Although the program plays a good game at the lowest level, it gets even stronger with extra thinking time. However, after 1 or 2 minutes per move diminishing returns set in and the increase in strength becomes proportionately less. This is because the program finds most of the plausible moves fairly quickly and then tends to spend the extra time choosing between them. When I threatened it with a crude mate in 1 it quickly found a defence which unfortunately allowed a more subtle mate in 3. It had to think for a further twenty minutes before discarding its original idea and finding the correct reply.

This does not detract from the fact that White Knight is a remarkable programming achievement. It plays chess that is sound and sensible even at its lowest and quickest levels. If you want a chess program for your Electron then this looks like the one to get.

SOLITAIRE

by Dave Robinson and Roy Walmsley

Solitaire is a computerised version of the one player game of the same name. Rather than use the normal pegs and board, this version makes good use of your Electron's graphics abilities.

The aim of the game is to finish with a single 'peg' in the centre of the board, the only position at the start that is unoccupied by a peg. Pegs are removed by hopping over them with other pegs. Moves can only be made vertically or horizontally - not diagonally.

The only legal move is to hop one peg over another into a hole. The peg hopped over is then removed.

Although this may sound easy, it is in fact very difficult! You tend to end up with several pegs out on a limb, with no hope of removing them.

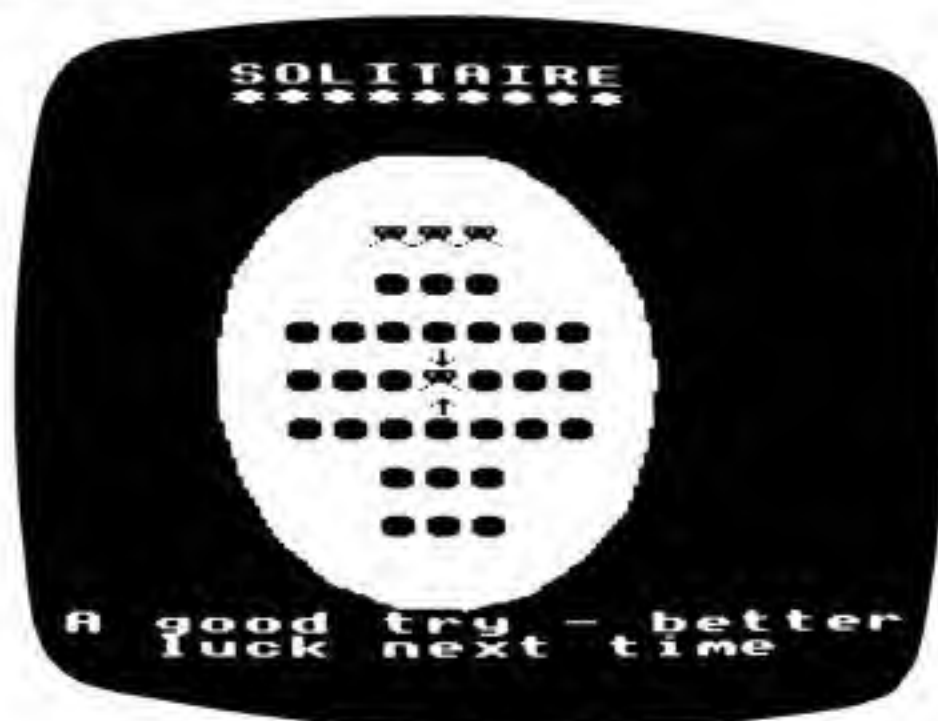
In this Electron version the pegs are in fact space invaders - all the more reason to remove them! You hop an invader by moving a cursor to it using the usual keys, and then hit the space bar. This 'picks up' the invader. You then move the cursor to the target hole and press the space bar again. If it is a legal move then the invader you have jumped over is banished forever to the very darkest corners of your Electron's memory.

If you pick up an invader and then change your mind about the whole move, you can reverse matters by simply returning it to its position and pressing the space bar again.

When you have won, or (shame) you have moved all that you are able, the program will detect this and finish with a suitable congratulatory or consolatory message.

PROGRAM NOTES

The bulk of the game is in, or called from PROCplay, at line 2020. This procedure looks after the movement keys and calls another procedure, PROCcursor, to move the cursor around the board. PROCcursor simply deletes the cursor in the old position, by



printing a space, and displays the cursor in the new position. There is also a check to see if you have tried to move the cursor off the edge of the cross-shaped playing area (line 1790).

The rest of PROCplay checks the state of the flag, up%. This reflects whether you are in the middle of a 'hop' or just moving around the board prior to one. Either of the procedures PROCpick and PROCputdown are called depending on this flag.

The procedure, PROCcheck is called each time that PROCplay is executed after a hop, to check if the game is over yet.

You'll never find a more peaceful way to zap the aliens! Good luck.

```
10 REM Program SOLITAIRE
20 REM Version El.0
30 REM Authors D.Robinson
   & R.Walmsley
40 REM ELBUG December 1984
50 REM Program subject to copyright
60 :
70 ON ERROR GOTO 2750
80 :
100 MODE 1
110 PROCinit
```



```

120 PROCinstruct
130 MODE 5
140 REPEAT
150 PROCreset
160 PROCdrawboard
170 REPEAT
180 PROCplay
190 UNTIL end%
200 IF board%(5,5)=226 AND peg%=1 THEN
N won%=TRUE
210 PROCending
220 UNTIL finish%
230 MODE6
240 END
250 :
1000 DEF PROCinit
1010 DIM board%(10,10)
1020 R%=310
1030 FOR I%=1 TO 10
1040 FOR J%=1 TO 10
1050 board%(I%,J%)=32
1060 NEXT:NEXT
1070 VDU23,224,0,8,28,42,8,8,8,0
1080 VDU23,226,126,90,126,60,36,66,129
,0
1090 VDU23,225,60,126,126,126,126,126,
60,0
1100 VDU23,227,0,8,8,8,42,28,8,0
1110 VDU23,1,0;0;0;0;0;
1120 ENDPROC
1130 :
1140 DEF PROCinstruct
1150 COLOUR 2
1160 PRINTTAB(15,1)"SOLITAIRE"
1170 PRINTTAB(15,2)STRING$(9,"*")
1180 COLOUR 3
1190 PRINTTAB(11,4)"Hop over the pegs"
1200 PRINTTAB(11,6)"to leave a single"
1210 PRINTTAB(11,8)"peg in the centre"
1220 COLOUR 2
1230 PRINTTAB(10,11)"To move the curso
r:"
1240 PRINTTAB(15,13)"Z - LEFT"
1250 PRINTTAB(15,15)"X - RIGHT"
1260 PRINTTAB(15,17)"* - UP"
1270 PRINTTAB(15,19)"/ - DOWN"
1280 PRINTTAB(10,23)"space bar to pick
up"
1290 PRINTTAB(11,25)"and put down a pe
g"
1300 COLOUR 3
1310 PRINTTAB(11,30)"SPACE BAR TO PLAY"
1320 REPEAT UNTIL GET=32
1330 ENDPROC
1340 :
1350 DEF PROCreset
1360 VDU23,1,0;0;0;0;0;
1370 FOR I%=2 TO 8
1380 FOR J%=4 TO 6
1390 board%(I%,J%)=226
1400 board%(J%,I%)=226
1410 NEXT:NEXT
1420 board%(5,5)=225
1430 peg%=32
1440 OX%=5:OY%=5
1450 end%=FALSE:finish%=FALSE
1460 won%=FALSE:up%=FALSE
1470 ENDPROC
1480 :
1490 DEF PROCdrawboard
1500 VDU19,0,4;0;
1510 VDU19,1,0;0;
1520 VDU19,2,1;0;
1530 VDU19,3,3;0;
1540 CLS
1550 COLOUR 3
1560 VDU 29,606;500;
1570 MOVE 0,0
1580 FOR A=0 TO 7 STEP 0.3
1590 MOVE 0,0:PLOT 85,R%*COS(A),R%*SIN
(A)
1600 NEXT A
1610 PRINTTAB(5,3)"SOLITAIRE"
1620 PRINTTAB(5,4)STRING$(9,"*")
1630 COLOUR 131
1640 FOR I%=2 TO 8
1650 FOR J%=2 TO 8
1660 PROCprint(board%(I%,J%),I%,J%)
1670 NEXT:NEXT
1680 PROCcursor(OX%,OY%)
1690 ENDPROC
1700 :
1710 DEF PROCprint(C%,X%,Y%)
1720 COLOUR C%MOD4
1730 PRINTTAB(X%+4,Y%*2+6)CHR$(C%)
1740 board%(X%,Y%)=C%
1750 ENDPROC
1760 :
1770 DEF PROCcursor(XC%,YC%)
1780 COLOUR 1
1790 IF board%(XC%,YC%)=32 THEN ENDPROC
1800 PRINTTAB(OX%+4,OY%*2+7)CHR$32
1810 PRINTTAB(OX%+4,OY%*2+5)CHR$32
1820 PRINTTAB(XC%+4,YC%*2+7)CHR$224
1830 PRINTTAB(XC%+4,YC%*2+5)CHR$227
1840 OX%=XC%:OY%=YC%
1850 ENDPROC
1860 :
1870 DEF PROCcheck
1880 IF peg%=1 THEN end%=TRUE:ENDPROC
1890 move%=FALSE
1900 FOR I%=1 TO 8
1910 FOR J%=1 TO 8
1920 IF board%(I%,J%)=226 THEN PROCgo
1930 IF move%=TRUE THEN I%=8:J%=8
1940 NEXT:NEXT
1950 IF move%=FALSE THEN end%=TRUE
1960 ENDPROC
1970 :
1980 DEF PROCgo

```



```

1990 IF board%(I%-1,J%)=226 AND board%
(I%-2,J%)=225 THEN move%=TRUE:ENDPROC
2000 IF board%(I%+1,J%)=226 AND board%
(I%+2,J%)=225 THEN move%=TRUE:ENDPROC
2010 IF board%(I%,J%-1)=226 AND board%
(I%,J%-2)=225 THEN move%=TRUE:ENDPROC
2020 IF board%(I%,J%+1)=226 AND board%
(I%,J%+2)=225 THEN move%=TRUE
2030 ENDPROC
2040 :
2050 DEF PROCplay
2060 IF up%=FALSE THEN PROCcheck
2070 IF end%=TRUE THEN ENDPROC
2080 REPEAT
2090 Xpos%=OX%:Ypos%=OY%
2100 A$=GET$
2110 IF A$="Z" THEN Xpos%=Xpos%-1
2120 IF A$="X" THEN Xpos%=Xpos%+1
2130 IF A$=":" THEN Ypos%=Ypos%-1
2140 IF A$="/" THEN Ypos%=Ypos%+1
2150 PROCcursor(Xpos%,Ypos%)
2160 UNTIL A$=CHR$32
2170 IF board%(Xpos%,Ypos%)=226 AND up
%=FALSE THEN PROCpick:ENDPROC
2180 IF board%(Xpos%,Ypos%)=225 AND up
%=TRUE THEN PROCputdown:ENDPROC
2190 IF board%(Xpos%,Ypos%)=227 AND up
%=TRUE THEN PROCreplace
2200 ENDPROC
2210 :
2220 DEF PROCpick
2230 Ifirst%=Xpos%
2240 Jfirst%=Ypos%
2250 SOUND 1,-15,80,3
2260 PROCprint(227,Xpos%,Ypos%)
2270 up%=TRUE
2280 ENDPROC
2290 :
2300 DEF PROCputdown
2310 Ilast%=Xpos%
2320 Jlast%=Ypos%
2330 Istep%=Ilast%-Ifirst%
2340 Jstep%=Jlast%-Jfirst%
2350 IF Istep%*Jstep%<>0 THEN ENDPROC
2360 IF ABS(Istep%+Jstep%)<>2 THEN END
PROC
2370 Imid%=Ifirst%+(Istep%DIV2)
2380 Jmid%=Jfirst%+(Jstep%DIV2)
2390 IF board%(Imid%,Jmid%)<>226 THEN
ENDPROC
2400 SOUND 1,-15,10,3
2410 PROCprint(225,Ifirst%,Jfirst%)
2420 SOUND 1,-15,80,3
2430 PROCprint(226,Ilst%,Jlast%)
2440 SOUND 1,-15,80,3
2450 PROCprint(225,Imid%,Jmid%)
2460 up%=FALSE
2470 peg%=peg%-1
2480 ENDPROC
2490 :
2500 DEF PROCreplace
2510 PROCprint(226,Xpos%,Ypos%)
2520 up%=FALSE
2530 ENDPROC
2540 :
2550 DEF PROCending
2560 T%=TIME+100:REPEAT UNTIL TIME>T%
2570 COLOUR 128:COLOUR2
2580 PRINTTAB(5,6)"Game Over"
2590 COLOUR3
2600 IF won% THEN PRINTTAB(2,26)"Congr
atulations"
2610 IF NOT(won%) THEN PRINTTAB(1,26)"
A good try - better"TAB(3,27)"luck next
time"
2620 COLOUR2
2630 FOR P%=200 TO 50 STEP-10
2640 SOUND 1,-15,P%,2
2650 NEXT P%
2660 T%=TIME+100:REPEAT UNTIL TIME>T%
2670 PRINTTAB(1,29)"Another game - Y/N"
2680 SOUND 1,-15,200,3
2690 REPEAT
2700 A%=INSTR("YyNn",GET$)
2710 UNTIL A%>0
2720 IF A%>2 THEN finish%=TRUE
2730 ENDPROC
2740 :
2750 ON ERROR OFF
2760 MODE 6
2770 IF ERR=17 THEN END
2780 REPORT:PRINT " at line ";ERL
2790 END

```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

DECIMALS TO FRACTIONS - Jonathan Wilson

The short program below takes as input a decimal number (eg 1.125), and outputs the lowest common denominator fraction (9/8 for the example above). Note that values of zero will produce the response 'Silly'.

```

10 INPUT"Decimal "D:T%=D*1E7:B%=1E7
20 REPEAT READC$:IF (T%MODC%=0) AND (B%MODC%=0) T%=T%/C%:B%=B%/C%:RESTORE
30 UNTIL C%=97:IF T%=0 PRINT "Silly" ELSE PRINT "Fraction ";T%"/";B%
40 DATA 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,97

```




FILE HANDLING (PART 1)

by Geoff Bains

Computers are ideally suited to storing large quantities of information. Although many Electron users want to make use of this ability of their micro, the whole subject is steeped in mystery. In this, the first of a series of articles on data files, Geoff Bains lifts the veil a little.

Everything that your Electron does it does by manipulating numbers. All text and strings are so many figures to the Electron (ASCII codes, in fact), all graphics are just a lot of co-ordinates. You know how to handle those subjects, however when it comes to storing and recalling these numbers themselves, many Electron owners flounder. Learning the methods of handling this data may be a new skill but the step involved is not as great as you'd think.

File handling is all about the storing and recalling of data - whether it is numeric or text. When data is manipulated within a program it is stored as a series of values. Using that system for the permanent storage of records is obviously not practical. For a start it is very wasteful of memory. In addition all values of variables are, of course, lost when you turn your Electron off. A permanent external method of storage is needed.

To store data we can use one of three methods:

- Data statements
- Cassette file
- Disc file

The last method is not yet available to Electron users as there is no disc interface available for the Electron. However, this will shortly change.

The first method uses the DATA statements that we know and love. Numbers stored in DATA statements can

be stored permanently by saving the program. The DATA statements are of course saved along with all the other Basic statements. The advantage of using DATA statements is that they provide very quick access to the data. Take a simple program like this, for example.

```
10 REM DATA statement demo
20 MODE 4
30 FOR I%=1 TO 12
40 READ month$,monthno
50 PRINT month$;" is month number ";
monthno
60 PRINT
70 NEXT I%
80 DATA January,1,February,2,March,3
90 DATA April,4,May,5,June,6
100 DATA July,7,August,8,September,9
110 DATA October,10,November,11,December,12
```

The program quickly reads through all the data and processes it - in this case it just prints it all out. Just printing out the data is none too useful. We want to make the computer work. The next program will search the data for a particular value.

```
10 REM DATA statement search demo
20 MODE 4
30 INPUT"Which month number do you want ",N
40 PRINT
50 FOR I%=1 TO 12
60 READ month$,monthno
70 IF monthno=N THEN PRINT "Month number ";N;" is ";month$
80 NEXT I%
90 DATA January,1,February,2,March,3
```


100 DATA April,4,May,5,June,6
 110 DATA July,7,August,8,September,9
 120 DATA October,10,November,11,December,12

Although this method does simple jobs such as these quite well it has one overriding disadvantage. To make any alterations or additions to the data you have to alter the program itself. This has to be done, as it were, manually, by editing. A Basic program cannot easily alter its own DATA statements. This drawback immediately rules out this method of storing data for any but the most trivial of applications.

Far better is the second method mentioned above. Storing data on cassette means that you can alter and add to the data with one program and process it with another, or even several other, programs.

Before we look at methods of handling cassette files it would be a good idea to establish a little of the (inevitable) jargon that accompanies this subject. For this we will consider an example - a library index file.

In libraries there is usually a card index containing details on all the books in the library. The entire index is called a 'file'. It is a group of related pieces of information. Each card in the index is a 'record'. The record contains all the information about one item (a book in this case). Each piece of information is called a 'field'. The author's name is a field, as is the British Library classification.

As a further example, a company might keep a 'file' of employee wages. This would contain several 'records', each concerning one employee. In each record there would be several 'fields', the employee's name, address, payroll number, wage, and so on. In this example the records are not on separate cards as the Library card index is, but they are distinct sections of the file and so are treated as entities in their own right.

To keep the early examples of cassette files simple we will start

with an example file that has only one field per record. That is, a simple list. Let's consider a list of home computers you may have considered before you bought your Electron.

Home computers

Amstrad	Electron
Atmos	Spectrum
BBC micro	VIC 20
CBM 64	ZX 81

We will write a program that produces a cassette file of this data. There are several Basic commands that are not used in any other applications that you will need to learn to handle data files.

Data handling commands.

BPUT#	OPENIN
BGET#	OPENOUT
CLOSE#	OPENUP
EOF#	PRINT#
INPUT#	PTR#

Using a cassette file is in many ways similar to using a file in an office. You do not need to know the details of the filing operations. The Electron looks after all that for you. It is as though there is a secretary to look after the filing system. The Electron can use up to five different secretaries, each looking after five different files, at once. However, with a cassette system, using more than one secretary at a time is impractical.

At the start of your work you must tell the secretary that you are going to be using a file. The OPEN commands, above, do this. To 'open' a file for output - writing data from the computer to a cassette file - we use OPENOUT:

CAROL=OPENOUT "MICROS"

We have told the imaginary secretary called CAROL to use a file called MICROS. When you type in this command you should have a cassette ready for recording in your recorder. As soon as you press Return after this statement the recorder will start and the header for the file is recorded onto the tape.

CAROL is the name of the secretary that we are to use. Of course the

Electron doesn't really use secretaries. It actually uses a system known as channels. We have opened a channel for output and called it CAROL. We could have called it anything we liked. The Electron just uses numbers to distinguish between channels. You don't need to know which channel is actually being used but if you type in PRINT CAROL, it'll tell you.

So far we haven't put anything in the file. We can do this with the command PRINT#. PRINT# is totally different from PRINT. You must not confuse the two. Now we can fill our cassette file with the names of the micros:

```
PRINT# CAROL,"Amstrad", "Atmos", "BBC micro", "CBM 64", "Electron", "Spectrum", "VIC 20", "ZX 81"
```

First the secretary (channel) is specified and then the data. When we have finished we must tell the secretary this and close the file. We do this with the command CLOSE#

```
CLOSE# CAROL
```

We can put all of this into a short program:

```
10 REM simple cassette file demo
20 CAROL=OPENOUT "MICROS"
30 PRINT# CAROL, "Amstrad", "Atmos", "BBC micro", "CBM 64", "Electron", "Spectrum", "VIC 20", "ZX 81"
40 CLOSE# CAROL
```

You will find that any error in line 30 will cause the program to stop with the file left open. It is not a good idea to leave open files lying around. So if this occurs you must close the file, as it were, by hand. Type in CLOSE# CAROL or even CLOSE#0, which closes any files that are open.

Now you have a file consisting of a list of micros. In the next issue we will see how this data can be read back into the Electron and some use made of it.

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

BASIC MEMORY RESERVE

When using the 6502 assembler built into Basic, you may wish to reserve an amount of memory for your program. This can be accomplished by using the EQU\$ pseudo command, like this:

```
.memory EQU$ STRING$(amount,CHR$0)
```

Note that no more than 255 bytes may be reserved like this. This method also ensures that the memory is cleared to a specific value (in this case 0).

MORE MEMORY - M.R. Bowers

If you are not using the cassette system for file access (not using OPENIN, OPENOUT or OPENUP), there is another 1/2k of memory available from &900 to &AFF. Note that loading and saving to cassette by Basic does not use this area anyway.

STRANGE ERROR LINES

Basic, although quite accurate at spotting errors, doesn't always manage to get the error line correct. For example, if you run the program below, it says that the error is in line 40, whereas it is line 10 that needs to be corrected! If you are writing a program, and find an error in the 'DEF PROC' part of a procedure, check the line that is calling it for errors as well. Another error (a bit more subtle) would have arisen if we'd just put A inside the brackets at line 10; the program would have worked, but printed up a zero. Strange!

```
10 PROCError(VAL 10)
20 END
30 :
40 DEF PROCError(A)
50 PRINT A
60 ENDPROC
```


GEORGE AND THE DRAGON

by O. R. Thomas

Now it's time for some light-hearted relief from zapping and blasting aliens. No sooner has your Christmas pudding settled than you are off to rescue 'Hideous Hilda' from the flames of the dragon, in this fast and exciting action game.

The idea of the game is to run along each level, jumping over the moving hole to collect the key which will allow George to climb the ladder up to the next level. George needs to reach the top level to free 'Hideous Hilda' before the dragon's flame reaches her. If you succeed in freeing Hilda, you proceed onto the next screen, where in addition to moving holes in the floor, you will have to face the additional hazard of arrows being fired just above your head.

The keys to use to play this game are 'Z' and 'X' for left and right, plus '/' to climb the ladders and 'Shift' to jump. There are nine different skill levels ranging from 1, which is fast, to 9 which is slow. As you complete each screen, the game automatically gets faster and faster.

Remember to take extra care when typing in the character and string definitions (between lines 1070 and

1330), as mistakes here will corrupt the screen display when you run the program.

So now you are ready to undertake your perilous quest, to thwart the wicked dragon by reaching 'Hideous Hilda' in the nick of time.

```

10 REM PROGRAM GEORGE
20 REM VERSION E0.1
30 REM AUTHOR O.R.THOMAS
40 REM ELBUG DECEMBER 1984
50 REM PROGRAM SUBJECT TO COPYRIGHT
60 :
100 ON ERROR GOTO 2890
110 MODE1
120 PROCInstructions
130 hi%=0:MODE 1:REPEAT
140 PROCskill:PROCvariables
150 FOR lives%=2 TO 0 STEP-1
160 MODE5:PROCcharacters
170 PROCscreen:PROCsetup
180 time%=0:totaltime%=0
190 REPEAT:time%=time%+1
200 IF time%>=pause% time%=0:totaltime%=totaltime%+1:VDU5:GCOLOR,2:MOVE(totaltime%*32)+32,816:PRINTflame$:VDU4
210 IF totaltime%=37 dead%=TRUE
220 PROCman
230 IF NOT jump% AND xpos%=holepos% dead%=TRUE:GOTO290
240 PROCChole
250 IF NOT jump% AND xpos%=holepos% dead%=TRUE:GOTO290
260 IF xpos%=keypos% AND ypos%=keyheight% key%=TRUE:SOUND1,-15,100,1:score%=score%+50:PROCscore:keypos%=20

```




```

270 IF arrow% AND xpos%=arrowpos% AND
ypos%=keyheight% dead%=TRUE:GOTO290
280 IF arrow% PROCarrow
290 UNTIL dead%:PROCdeath
300 FOR ag%=1 TO 15:SOUND0,-15+ag%,10
0,2:NEXT
310 jump%=FALSE:jumpno%=0:ychange%=0:
xchange%=0
320 dead%=FALSE:PROCscore:NEXT
330 COLOUR2:PRINTTAB(4,17)STRING$(12,
CHR$32);TAB(4,18)" GAME OVER ";TAB(4,1
9)STRING$(12,CHR$32)
340 TIME=0:REPEAT UNTIL TIME>300
350 IF score%>hi% hi%=score%
360 MODEL:PROCscodisp
370 UNTIL Q$="N":MODE 6
380 END
390 :
1000 DEF PROCvariables
1010 key%=FALSE:dead%=FALSE:score%=0
1020 level%=1:jump%=FALSE:jumpno%=0
1030 arrow%=FALSE:arrowpos%=10
1040 ENDPROC
1050 :
1060 DEF PROCcharacters
1070 VDU23,224,56,56,56,48,62,62,56,56
:VDU23,225,56,56,56,48,48,48,48,56
1080 VDU23,226,28,28,28,12,124,124,28,
28:VDU23,227,28,28,28,12,12,12,12,28
1090 VDU23,228,60,60,189,153,255,255,6
0,60:VDU23,229,60,60,60,36,36,36,36,102
1100 VDU23,230,0,0,0,0,24,44,118,175
1110 VDU23,231,32,32,48,60,251,255,254
,252:VDU23,232,248,252,255,254,224,96,1
12,80
1120 VDU23,233,0,16,32,96,224,96,32,16
:VDU23,234,60,60,126,126,255,255,36,102
1130 VDU23,235,96,97,149,159,159,144,9
6,96:VDU23,236,255,255,255,129,66,36,24
,255
1140 VDU23,237,129,129,129,255,255,129
,129,129:VDU23,238,0,0,0,99,254,99,0,0
1150 VDU23,239,0,0,0,198,127,198,0,0:VD
U23,240,129,129,129,129,129,129,129,129
1160 VDU19,3,4;0;
1170 VDU23,1,0;0;0;0;
1180 right$=CHR$17+CHR$1+CHR$224+CHR$1
0+CHR$8+CHR$17+CHR$2+CHR$225
1190 left$=CHR$17+CHR$1+CHR$226+CHR$10
+CHR$8+CHR$17+CHR$2+CHR$227
1200 still$=CHR$17+CHR$1+CHR$228+CHR$1
0+CHR$8+CHR$17+CHR$2+CHR$229
1210 dead$=CHR$32+CHR$10+CHR$8+CHR$17+
CHR$2+CHR$230
1220 dragon$=CHR$17+CHR$1+CHR$231+CHR$
10+CHR$8+CHR$232
1230 flame$=CHR$17+CHR$2+CHR$233
1240 maiden$=CHR$17+CHR$2+CHR$228+CHR$
10+CHR$8+CHR$17+CHR$1+CHR$234
1250 key$=CHR$17+CHR$3+CHR$235

```

GEORGE AND THE DRAGON
by O.R.Thomas

Help George to dodge arrows and leap over holes as he rushes to grasp the keys allowing him further up the battlements. But hurry - you must rescue Hideous Hilda before the dragon's flames reach her.

The controls are as follows :

PRESS SPACE TO CONTINUE

```

1260 safefloor$=CHR$17+CHR$1+CHR$236
1270 dangerfloor$=CHR$17+CHR$3+CHR$236
1280 ladder$=CHR$17+CHR$2+CHR$237
1290 doubleladder$=CHR$17+CHR$2+CHR$23
7+CHR$10+CHR$8+CHR$237
1300 arrowleft$=CHR$17+CHR$2+CHR$238
1310 arrowright$=CHR$17+CHR$2+CHR$239
1320 gap$=CHR$17+CHR$1+CHR$240
1330 blank$=CHR$32+CHR$10+CHR$8+CHR$32
1340 ENDPROC
1350 :
1360 DEF PROCscreen
1370 GCOLOR,130:VDU24,0;847;1279;1023;
1380 CLG:VDU24,0;16;1279;80;:CLG
1390 GCOLOR,128:VDU24,8;855;1271;1015;
1400 CLG:VDU24,8;24;1271;72;:CLG
1410 VDU24,0;0;1279;1023;:VDU5
1420 GCOLOR,1:MOVE24,991
1430 PRINT"George & The Dragon"
1440 GCOLOR,2:MOVE32,995
1450 PRINT"George & The Dragon"
1460 VDU4:PROCscore
1470 PRINTTAB(6,3)"SCORE : "
1480 PRINTTAB(9,4)"HI : "
1490 PRINTTAB(6,30)"LEVEL ";level%
1500 PRINTTAB(0,6)dragon$
1510 PRINTTAB(19,6)maiden$
1520 PRINTTAB(1,26)still$
1530 xpos%=1:ypos%=26
1540 PRINTTAB(0,8)STRING$(20,dangerflo
or$)
1550 FOR ah%=12 TO 28 STEP 4
1560 PRINTTAB(0,ah%)STRING$(20,safeflo
or$)
1570 PRINTTAB(4,ah%)STRING$(12,dangerf
loor$)
1580 NEXT:anxx%=18
1590 FOR ah%=8 TO 24 STEP 4
1600 FOR anyy%=ah% TO ah%+3
1610 PRINTTAB(anxx%,anyy%)ladder$
1620 NEXT
1630 IF anxx%=18 anxx%=1 ELSE anxx%=18
1640 NEXT
1650 ENDPROC

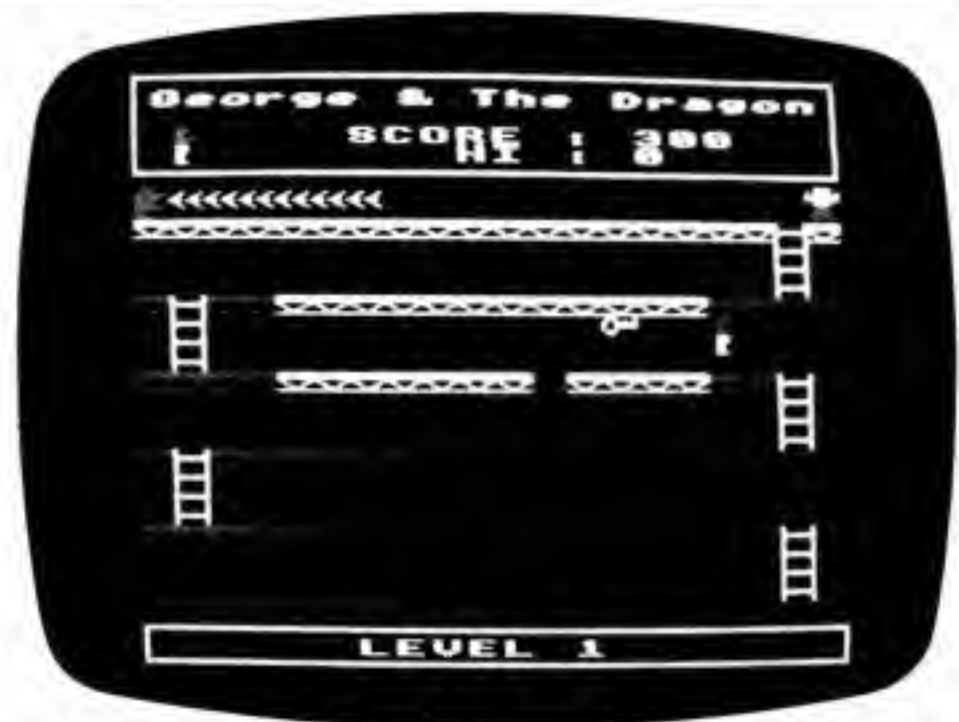
```



```

1660 :
1670 DEF PROCscore
1680 COLOUR2
1690 IF lives%<1 GOTO 1730
1700 FOR ah%=0 TO lives%-1
1710 PRINTTAB(ah%+1,3)right$
1720 NEXT
1730 PRINTTAB(lives%+2,3)blank$
1740 PRINTTAB(14,3);score%
1750 PRINTTAB(14,4);hi%
1760 ENDPROC
1770 :
1780 DEF PROCman
1790 IF jump% PROCjump:GOTO1860
1800 xchange%=0
1810 ychange%=0
1820 IF INKEY(-98) AND NOT jump% xchan
ge%=-1
1830 IF INKEY(-67) AND NOT jump% xchan
ge%=1
1840 IF INKEY(-1) jump%=TRUE
1850 IF INKEY(-105) AND POINT(xpos%*64
,(32-ypos%)*32+36)=2 AND key% PROCclimb
1860 IF xchange%<>0 OR jump% PRINTTAB(
xpos%,ypos%)blank$:IF POINT(xpos%*64,(3
2-ypos%)*32+36)=2 AND NOT jump% OR POIN
T(xpos%*64,(32-ypos%)*32+4)=2 PRINTTAB(
xpos%,ypos%)doubleladder$
1870 IF jumpno%=3:jumpno%=0:jump%=FALSE
1880 xpos%=xpos%+xchange%
1890 ypos%=ypos%+ychange%
1900 IF xpos%<0 xpos%=0
1910 IF xpos%>19 xpos%=19
1920 IF xchange%=1 PRINTTAB(xpos%,ypos
%)right$
1930 IF xchange%=0 PRINTTAB(xpos%,ypos
%)still$
1940 IF xchange%=-1 PRINTTAB(xpos%,ypo
s%)left$
1950 ENDPROC
1960 :
1970 DEF PROCclimb
1980 key%=FALSE
1990 FOR ah%=1 TO 4
2000 SOUND1,-15,50*ah%,2
2010 IF ah%=1 PRINTTAB(xpos%,ypos%+1)l
adder$:GOTO2030
2020 PRINTTAB(xpos%,ypos%+1)doubleladd
er$
2030 ypos%=ypos%-1
2040 PRINTTAB(xpos%,ypos%)still$
2050 NEXT
2060 score%=score%+10:PROCscore
2070 IF ypos%=6 CLS:level%=level%+1:sc
ore%=score%+((37-totaltime%)*10):PROCsc
reen:totaltime%=0:IF pause%>5 pause%=pa
use%-1
2080 PRINTTAB(arrowpos%,keyheight%);CH
R$32

```



```

2090 IF POINT(arrowpos%*64,(32-keyheig
ht%)*32+4)=2 PRINTTAB(arrowpos%,keyheig
ht%)ladder$
2100 PROCsetup
2110 ENDPROC
2120 :
2130 DEF PROCjump
2140 jumpno%=jumpno%+1
2150 IF jumpno%=1 ychange%=-1
2160 IF jumpno%=2 ychange%=0
2170 IF jumpno%=3 ychange%=1
2180 SOUND1,-10,140+(30*ychange%),1
2190 ENDPROC
2200 :
2210 DEF PROCsetup
2220 IF ypos%<26 PRINTTAB(4,ypos%+6)ST
RING$(12,safefloor$)
2230 holepos%=10:holedir%=1
2240 keypos%=RND(12)+3:keyheight%=ypos
%-1
2250 key%=FALSE:PRINTTAB(keypos%,ypos%
-1)key$
2260 IF level%>1 arrow%=TRUE:arrowpos%
=RND(18):IF arrowpos%<10 dir%=1 ELSE di
r%=-1
2270 ENDPROC
2280 :
2290 DEF PROCarrow
2300 PRINTTAB(arrowpos%,keyheight%);CH
R$32
2310 IF arrowpos%=keypos% PRINTTAB(arr
owpos%,keyheight%);key$
2320 IF POINT(arrowpos%*64,(32-keyheig
ht%)*32+4)=2 PRINTTAB(arrowpos%,keyheig
ht%)ladder$
2330 arrowpos%=arrowpos%+dir%
2340 IF dir%=1 AND arrowpos%>19 dir%=-
1:arrowpos%=19
2350 IF dir%=-1 AND arrowpos%<0 dir%=1
:arrowpos%=0
2360 IF dir%=1 PRINTTAB(arrowpos%,keyh
eight%)arrowright$
2370 IF dir%=-1 PRINTTAB(arrowpos%,key
height%)arrowleft$

```



```

2380 ENDPROC
2390 :
2400 DEF PROCchole
2410 PRINTTAB(holepos%,keyheight%+3)da
ngerfloor$
2420 holepos%=holepos%+holedir%
2430 IF holepos%>15 holepos%=15:holedi
r%=-1
2440 IF holepos%<4 holepos%=4:holedir%
=1
2450 PRINTTAB(holepos%,keyheight%+3)ga
p$
2460 ENDPROC
2470 :
2480 DEF PROCdeath
2490 IF totaltime%=37 PRINTTAB(19,6)de
ad$:GOTO 2550
2500 REPEAT
2510 PRINTTAB(xpos%,ypos%)CHR$32
2520 ypos%=ypos%+1
2530 PRINTTAB(xpos%,ypos%)still$
2540 UNTIL POINT(xpos%*64+32,(32-(ypos
%+2))*32-4)>0 OR ypos%=27
2550 PRINTTAB(xpos%,ypos%)dead$
2560 ENDPROC
2570 :
2580 DEF PROCinstructions
2590 CLS
2600 VDU23,1,0;0;0;0;0;:COLOUR1
2610 PRINT'TAB(10)"GEORGE AND THE DRAG
ON"
2620 COLOUR3:PRINTTAB(14)"by O.R.Thoma
s""
2630 COLOUR2:PRINT" Help George to do
dge arrows and leap""over holes as
he rushes to grasp the""keys allow
ing him further up the""battlem
ents. But hurry - you must";
2640 PRINT'"rescue Hideous Hilda
before the""dragon's flames reach
her."
2650 PRINT'"TAB(5)"The controls are a
s follows :":COLOUR1:PRINTTAB(14)"Z -
left""TAB(14)"X - right""TAB(14)
"/ - climb""TAB(10)"SHIFT - jum
p"
2660 COLOUR3:PRINTTAB(9,30)"PRESS SPAC
E TO CONTINUE";
2670 *FX15,1
2680 ah%=GET
2690 IF ah%<>32 GOTO 2680
2700 ENDPROC
2710 :
2720 DEF PROCskill
2730 VDU23,1,0;0;0;0;0;
2740 COLOUR2:PRINTTAB(11,14)"SKILL LEV
EL (1-9):"
2750 *FX15,1
2760 pause%=5+(GET-48)
2770 IF pause%<6 OR pause%>14 GOTO 2760
2780 COLOUR1:PRINTTAB(29,14);pause%-5
2790 TIME=0:REPEAT UNTIL TIME>70
2800 ENDPROC
2810 DEFPROCscodisp
2820 PRINTTAB(11,3)"YOUR SCORE WAS ";S
TR$(score%)
2830 PRINT'':
2840 COLOUR1:PRINTTAB(11)"ANOTHER GAME
?";
2850 *FX15,1
2860 Q$=GET$:PRINTQ$
2870 ENDPROC
2880 :
2890 ON ERROR OFF:MODE 6
2900 IF ERR=17 END
2910 REPORT:PRINT" at line ";ERL:END

```

HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS HINTS

BETTER INPUT

If you are using INPUT in one of your programs, and the user is likely to input some text with a comma in (","), then by using INPUT LINE, you can ensure that the whole string is input (including commas, quotes and leading spaces). If you don't use this, then only the string up to the comma will be used.

NEW WAY TO CRASH AN ELECTRON? - M.P. Briggs

Try DIM A\$(1,1,1,1,1,1,1,1,1,1,1,1,1,1) (there's 14 of them) to crash your Electron.

STRANGE VARIABLES - Tony Walsh

If you have BEEBUGSOFT's new Toolkit ROM for the Electron and you wish to make a program unalterable, then try changing variables to built in words, such as PRINT (with the search and replace option), but omit the £ sign to indicate that it is a Basic keyword. Toolkit then replaces your variable for one spelt like the Basic keyword, but not tokenised. Most attempts to change the program will result in the variable being tokenised, and thus Basic producing an error when the program is run.

BACK ISSUES AND SUBSCRIPTIONS

BACK ISSUES (Members only)

All back issues will be kept in print (from November 1983). Send 90p per issue PLUS an A5 SAE to the subscriptions address. Back copies of BEEBUG are available to ELBUG members at this same price. This offer is for members only, so it is ESSENTIAL to quote your membership number with your order. Please note that the advertising supplements are not supplied with back issues.

Subscription and Software Address

ELBUG
PO BOX 109
High Wycombe
Bucks

SUBSCRIPTIONS

Send all applications for membership, and subscription queries to the subscriptions address.

MEMBERSHIP COSTS:

U.K.

£5.90 for 6 months (5 issues)

£9.90 for 1 year (10 issues)

Eire and Europe

Membership £16 for one year.

Middle East £19

Americas and Africa £21

Elsewhere £23

Payments in Sterling preferred.

SOFTWARE (Members only)

This is available from the software address.

MAGAZINE CONTRIBUTIONS AND TECHNICAL QUERIES

Please send all contributions and technical queries to the editorial address opposite. All contributions published in the magazine will be paid for at the rate of £25 per page.

We will also pay £10 for the best Hint or Tip that we publish, and £5 to the next best. Please send all editorial material to the editorial address opposite. If you require a reply it is essential to quote your membership number and enclose an SAE.

Editorial Address

ELBUG
PO Box 50
St Albans
Herts

ELBUG MAGAZINE is produced by BEEBUG Publications Ltd.

Editor: Mike Williams.

Assistant Editor: Geoff Bains. Production Editor: Phyllida Vanstone.

Technical Assistants: David Fell and Alan Webster.

Managing Editor: Lee Calcraft.

Thanks are due to, Sheridan Williams, and Adrian Calcraft for assistance with this issue.

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility, whatsoever, for errors in articles, programs, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, BEEBUG Publications Limited.
BEEBUG Publications LTD (c) 1984.

New Elbug Binders

We have produced an attractive hard-backed binder for the ELBUG magazine. These binders are green in colour with "ELBUG" in gold lettering on the spine and allow for the whole of one volume of the magazine to be stored as a single reference book.

Each binder will accommodate 10 ELBUG magazines, and is supplied with 12 wires to enable the index and the latest copy of the supplement to be included within the binder if required. Individual issues may be easily added and removed, allowing for the latest volume to be filed as it arrives.



The price of the new ELBUG binder is £3.90 including VAT, please add 50p post and packing for delivery within the U.K. Overseas members please send the same amount, this will cover the extra postage but not VAT.

Please send to:

BEEBUGSOFT, PO BOX 109, High Wycombe, Bucks, HP10 8HQ.

THE BEST OF ELBUG ON CASSETTE

Many of the best programs published in ELBUG have been collected together and published by Penguin Books under the name "Games and other programs for the Acorn Electron" at £3.95. This book is part of the Penguin Acorn Computer Library and at present there is just one other title available though others are planned.

There are 20 programs in all in four different categories:

Action Games

Munch-Man	Mars Lander	Invasion
Robot Attack	Hedgehog	

Thought games

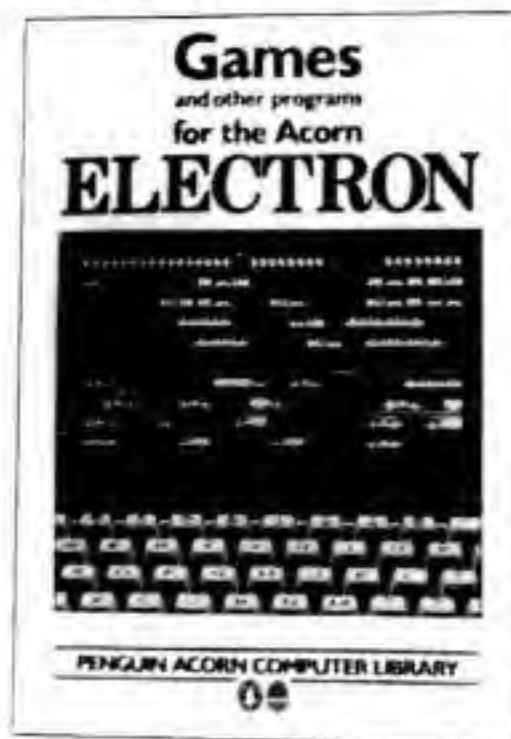
Higher/Lower	Five-Dice	Life
Anagrams	Return of the Diamond	

Visual Displays

Union Jack	Square Dance	Ellipto
Screenplay	3-D Rotation	

Utilities

Sound Wizard	Bad Program Lister
3-D Lettering	Bad Program Rescue
Double Height Text	



All 20 programs are now available on cassette from our software address (in High Wycombe) price £7 to members and £9 to non-members, plus 50p post & packing in either case.

ELBUG MAGAZINE CASSETTE

To save wear and tear on fingers and brain, we offer, each month, a cassette of the programs featured in the latest edition of ELBUG. The first program on each tape is a menu program, detailing the tape's contents, and allowing the selection of individual programs. The tapes are produced to a high technical standard by the process used for the BEEBUGSOFT range of titles.

Magazine cassettes have been produced for each issue of ELBUG from Volume 1 Number 1 onwards and are all available from stock, priced £3.00 each inclusive of VAT. See below for ordering information.

This month's cassette includes:

Volume 2 Number 1

A Christmas Carol, an exciting Power Boats race game, three short games – Treasure Hunt, Asterisk Tracker and Truffle Hunt, a useful routine for producing proportionally spaced text in mode 5, a Solitaire game but with a computing theme, a fascinating Zoom graphics program together with the data for our Zoom Competition, and a superb original game called George and the Dragon.

MAGAZINE CASSETTE SUBSCRIPTION

We are also able to offer ELBUG members subscription to the magazine cassette, this gives the added advantage of receiving the cassette at around the same time as the magazine each month. Subscriptions may either be for a period of 1 year or 6 months. (NOTE Magazine cassettes are produced 10 times each year).

If required, subscriptions may be backdated as far as Volume 1 Number 1, so when applying please write to the address below quoting your membership number and the issue from which you would like your subscription to start.

MAGAZINE CASSETTE ORDERING INFORMATION

Individual ELBUG Magazine Cassettes £3.00.

P & P: Please add 50p for the first and 30p for each subsequent cassette.

Overseas orders: Please send the same amount, this will include the extra post but not VAT.

Magazine Cassette Subscription

1 YEAR (10 issues)	£33.00 Inclusive O'SEAS	£39.00 No VAT payable
6 MONTHS (5 issues)	£17.00 Inclusive O'SEAS	£20.00 No VAT payable

Please be sure to specify that you require subscription to the ELBUG magazine cassette (as opposed to the BEEBUG cassette), and enclose your membership number with a cheque made payable to BEEBUGSOFT.

Please send to . .

ELBUG Magazine Cassette, BEEBUGSOFT, PO Box 109, High Wycombe, HP10 8HQ